IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

FINJAN SOFTWARE, LTD., an Israel corporation, Plaintiff-Counterdefendant, v. SECURE COMPUTING CORPORATION, a Delaware corporation, CYBERGUARD, CORPORATION, a Delaware corporation, WEBWASHER AG, a German corporation and DOES 1 THROUGH 100, Defendants-Counterclaimants.	))))))))))))	C. A. No. 06-369 (GMS) **PUBLIC VERSION**

JOINT APPENDIX OF INTRINSIC AND EXTRINSIC EVIDENCE RE: CLAIM CONSTRUCTION BRIEFING

VOLUME 2

OF COUNSEL:

Paul J. Andre
Lisa Kobialka
Meghan A. Wharton
James R. Hannah
Perkins Coie LLP
101 Jefferson Drive
Menlo Park, CA 94025-1114
(650) 838-4300

Dated: September 28, 2007
Public Version: October 4, 2007

Philip A. Rovner (#3215)
POTTER ANDERSON & CORROON LLP
Hercules Plaza
P. O. Box 951
Wilmington, DE 19899
(303) 984-6000
provner@potteranderson.com

Attorneys for Plaintiff
Finjan Software, Ltd.

JOINT APPENDIX OF INTRINSIC AND EXTRINSIC EVIDENCE

Tab	Description	Party Citing	Page(s)
1	U.S. Patent No. 6,092,194	Finjan Secure Computing	JA1 – JA20
2	U.S. Patent No. 6,804,780	Finjan Secure Computing	JA21 - JA38
3	U.S. Patent No. 7,058,822	Finjan Secure Computing	JA39 - JA62
4	U.S. Patent No. 6,357,010	Finjan Secure Computing	JA63 – JA83
5	U.S. Patent No. 7,185,361	Finjan Secure Computing	JA84 - JA93
6	'194 Patent, Office Action dated January 7, 1999	Finjan	JA181 - JA190
7	'194 Patent, Office Action mailed June 7, 1999	Finjan	JA207 – JA215
8	'194 Patent, Response to Office Action, dated October 27, 1999	Finjan	JA228 - JA234
9	'194 Patent, Information Disclosure Citation	Finjan	JA250
10	'780 Patent, Response to Office Action, dated July 31, 2003	Finjan	JA404 – JA412
11	'010 Patent, Response to Office Action, dated September 19, 2000	Finjan	JA926 - JA947

12	'361 Patent, Office Action mailed September 10, 2003	Finjan	JA1087 – JA1098
13	'361 Patent, Response to Office Action, dated January 12, 2004	Finjan	JA1100 – JA1111
14	'361 Patent, Final Rejection/Office Action, mailed on February 18, 2004	Finjan	JA1113 – JA1124
15	'361 Patent, Appellants' Brief on Appeal Under 37 C.F.R.41.37©	Finjan	JA1130 – JA1155
16	Zhang, X.N. *Secure Code Distribution.* June, 1997.	Finjan Secure Computing	JA1228 - JA1232
17	Excerpts from the Deposition of Martin Stecher, Augsut 28 & 29, 2007	Finjan	JA1233 – JA1239
18	Excerpts from the Deposition of Steven O. Chew, September 7, 2007	Finjan	JA1240 – JA1246
19	Excerpts from the Deposition of Frank Berzau, August 31, 2007	Finjan	JA1247 - JA1250
20	Excerpts from the Deposition of Christoph Alme, August 29, 2007	Finjan	JA1251 - JA1258
21	Excerpts from the Deposition of Roland Scholz, August 31, 2007	Finjan	JA1259 – JA1264
22	Excerpts from the Deposition of Michael J. Gallagher, August 3, 2007	Finjan	JA1265 – JA1269
23	Excerpts from the Deposition of Peter Borgolte, August 31, 2007	Finjan	JA1270 – JA1274
24	Excerpts from the Deposition of Jan Schnellbacher, August 31, 2007	Finjan	JA1275 - JA1281
25	U.S. Patent No. 6,167,520	Secure Computing	JA2000 - JA2012
26	Issue Notification to the '361 Patent	Secure Computing	JA2013
27	'361 Patent, Office Action mailed September 10, 2003	Secure Computing	JA2014 - JA2024

28	'361 Patent, January 12, 2004 Response to Office Action	Secure Computing	JA2025 - JA2036
29	'010 Patent, Cover of U.S. Utility Patent Application No. 09/024,576	Secure Computing	JA2037
30	'010 Patent, July 17, 2001 Response to Office Action	Secure Computing	JA2038 – JA2044
31	'194 Patent, Cover of U.S. Utility Patent Application No. 08/964,388	Secure Computing	JA2045
32	'194 Patent, October 27, 1999 Response to Office Action	Secure Computing	JA2046 - JA2052
33	'780 Patent, Cover to U.S. Utility Patent Application No. 09/539,667	Secure Computing	JA2053
34	'780 Patent, July 31, 2003 Response to Office Action	Finjan Secure Computing	JA2054 – JA2062
35	Letter from Forrester & Boehmert to European Patent Office re: European Patent Application No. 97950351.3 (December 21, 2005)	Finjan Secure Computing	JA2063 - JA2067
36	McDaniel, George. *IBM Dictionary of Computing*. (Tenth Edition, 1994)	Secure Computing	JA2068 – JA2072
37	Dowining, Douglas A., Covington, Michael A., & Covington. *Dictionary of Computer and Internet Terms*. (Fifth Edition, 1996)	Secure Computing	JA2073 - JA2076
38	*Dictionary of Computer Words: An A to Z Guide to Today's Computers*. (Revised Edition, 1995)	Secure Computing	JA2077 - JA2079
39	*Epicrealm, Licensing, LLC v. Autoflex Leasing, Inc., et al*. 2006 WL 3099603 (E.D.Tex.)	Secure Computing	JA2080 - JA2095
40	*Collegenet, Inc. v. XAP Corp*. 2004 WL 2429843 (D.Or.)	Secure Computing	JA2096 - JA2132
41	*Pipe Liners, Inc. v. Pipelining Products, Inc.* 1999 WL 1011974 (D.Del.)	Secure Computing	JA2133 – JA2146
42	Excerpts from Deposition transcript of Yuval Ben-Itzhak, August 10, 2007	Secure Computing	JA2147 – JA2152

43	Excerpts from Deposition transcript of Michael J. Gallagher, August 3, 2007	Secure Computing	JA2153 – JA2155
44	Excerpts from Rough Transcript of Deposition of Martin Stecher, August 28, 2007	Secure Computing	JA2156 – JA2158
45	Chappell, David. *Introducing Active X.* (January 1997).	Secure Computing	JA2159 – JA2162
46	Search result from USPTO Trademark Electronic Search System (TESS) on September 7, 2007	Secure Computing	JA2163 – JA2164
47	'361 Patent, Appellants' Brief on Appeal	Secure Computing	JA2165 - JA2190
48	Finjan Software Ltd. – Financial Statements as of December 31, 1998	Secure Computing	JA2191- JA2206
49	Finjan Software Ltd. and Its Subsidiary – Consolidated Financial Statements as of December 31, 1999	Secure Computing	JA2207 - JA2222
50	Finjan Software Ltd.– Consolidated Financial Statements as of December 31, 2000	Secure Computing	JA2223 - JA2243
51	Finjan Software Ltd. and Its Subsidiary – Consolidated Financial Statements as of December 31, 2001	Secure Computing	JA2244 – JA2262
52	Finjan Software, Inc., and Its Subsidiaries – Consolidated Financial Statements as of December 31, 2002	Secure Computing	JA2263 -2281
53	Finjan Software, Inc. and Its Subsidiaries – Consolidated Financial Statements as of December 31, 2003	Secure Computing	JA2282 – 2304
54	Finjan Software, Inc. and Its Subsidiaries – Consolidated Financial Statements as of December 31, 2004	Secure Computing	JA2305 – JA2325
55	Finjan Software Ltd. and Its Subsidiary – Consolidated Financial Statements dated December 31, 2005	Secure Computing	JA2326- JA2347
56	Webster's New World Dictionary of Computer Terms (Fifth Edition)	Secure Computing	JA2348 - JA2350

57	Garner, Bryan A. *Modern American Usage.* Oxford University Press, 2003.	Secure Computing	JA2351-JA2353
58	Secure Computing – Company Fact Sheet	Secure Computing	JA2354 - JA2360
59	Secure Computing – "Sidewinder – The Origin of Sidewinder G2 Security Appliance."	Secure Computing	JA2361 - JA2364
60	U.S. Patent No. 5,864,683	Secure Computing	JA2365 - JA2401
61	Committee Membership Information: Technical Privacy Dimensions of Information for Terrorism Prevention and Other National Goals	Secure Computing	JA2402 - JA2413

IN THE UNITED STATES DISTRICT COURT FOR THE DISTRICT OF DELAWARE

CERTIFICATE OF SERVICE

I, Philip A. Rovner, hereby certify that on October 4, 2007, the within document was filed with the Clerk of the Court using CM/ECF which will send notification of such filing(s) to the following; that the document was served on the following counsel as indicated; and that the document is available for viewing and downloading from CM/ECF.

BY HAND DELIVERY

Frederick L. Cottrell, III, Esq.
Kelly E. Farnan, Esq.
Richards, Layton & Finger, P.A.
One Rodney Square
920 N. King Street
Wilmington, DE 19801
cottrell@rlf.com; farnan@rlf.com

I hereby certify that on October 4, 2007 I have sent by Federal Express the foregoing document to the following non-registered participants:

Jake M. Holdreith, Esq.
Christopher A. Seidl, Esq.
Robins, Kaplan, Miller & Ciresi L.L.P.
2800 LaSalle Plaza
800 LaSalle Avenue
Minneapolis, MN 55402
jmholdreith@rkmc.com ; caseidl@rkmc.com

/s/ Philip A. Rovner
Philip A. Rovner (#3215)
Potter Anderson & Corroon LLP
Hercules Plaza
P.O. Box 951
Wilmington, Delaware 19899
(302) 984-6000
E-mail: provner@potteranderson.com

US006167520A

United States Patent [19]

Touboul

[11] **Patent Number: 6,167,520**

[45] **Date of Patent: Dec. 26, 2000**

[54] **SYSTEM AND METHOD FOR PROTECTING A CLIENT DURING RUNTIME FROM HOSTILE DOWNLOADABLES**

[75] Inventor: **Shlomo Touboul**, Kefar-Haim, Israel

[73] Assignee: **Finjan Software, Inc.**, San Jose, Calif.

[21] Appl. No.: **08/790,097**

[22] Filed: **Jan. 29, 1997**

Related U.S. Application Data

[60] Provisional application No. 60/030,639, Nov. 8, 1996.

[51] **Int. Cl.**[7] **G06F 11/30**; H04L 9/00

[52] **U.S. Cl.** **713/200**; 709/225

[58] **Field of Search** 395/186, 200.55, 395/200.59; 364/222.5, 286.4, 286.5; 326/8; 711/163; 713/200, 201; 380/4, 25

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,077,677	12/1991	Murphy et al.	395/10
5,359,659	10/1994	Rosenthal	380/4
5,361,359	11/1994	Tajalli et al.	395/700
5,485,409	1/1996	Gupta et al.	395/186
5,485,575	1/1996	Chess et al.	395/183.14
5,572,643	11/1996	Judson	395/793
5,623,600	4/1997	Ji et al.	395/187.01
5,638,446	6/1997	Rubin	380/25
5,692,047	11/1997	McManis	380/4
5,692,124	11/1997	Holden et al.	395/187.01
5,720,033	2/1998	Deo	395/186
5,724,425	3/1998	Chang et al.	380/25
5,740,248	4/1998	Fieres et al.	380/25
5,761,421	6/1998	Van Hoff et al.	395/200.53
5,765,205	6/1998	Breslau et al.	711/203
5,784,459	7/1998	Devarakonda et al.	380/4
5,796,952	8/1998	Davis et al.	395/200.54
5,805,829	9/1998	Cohen et al.	395/200.32
5,832,208	11/1998	Chen et al.	395/187.01
5,850,559	12/1998	Angelo et al.	395/750.03
5,859,966	1/1999	Hayman et al.	395/186
5,864,683	1/1999	Boebert et al.	395/200.79
5,892,904	4/1999	Atkinson et al.	395/187.01
5,956,481	9/1999	Walsh et al.	395/186
5,983,348	11/1999	Ji	713/200

OTHER PUBLICATIONS

IBM AntiVirus User's Guide Version 2.4, p. 6–7, Nov. 1995.

Zhang, X.N., *Computer*, "Secure Code Distribution," vol. 30, Jun., 1997, pp.: 76–79.

"Finjan Announces a Personal Java™ Firewall For Web Browsers—the SurfinShield™ 1.6", Press Release of Finjan Releases SurfinShield, Oct. 21, 1996, 2 pages.

"Finjan Software Releases SurfinBoard, Industry's First JAVA Security Product For the World Wide Web", Article published on the Internet by Finjan Software, Ltd., Jul. 29, 1996, 1 page.

"Powerful PC Security for the New World of Java™ and Downloadables, Surfin Shield™" Article published on the Internet by Finjan Software Ltd., 1996, 2 Pages.

"Company Profile Finjan—Safe Surfing, The Java Security Solutions Provider" Article published on the Internet by Finjan Software Ltd., Oct. 31, 1996, 3 pages.

"Finjan Announces Major Power Boost and New Features for SurfinShield™ 2.0" Las Vegas Convention Center/Pavillion 5 P5551, Nov. 18, 1996, 3 pages.

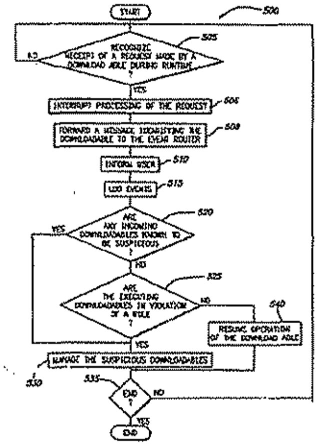
(List continued on next page.)

Primary Examiner—Dieu-Minh T. Le
Attorney, Agent, or Firm—Graham & James LLP

[57] **ABSTRACT**

A system and method examine execution or interpretation of a Downloadable for operations deemed suspicious or hostile, and respond accordingly. The system includes security rules defining suspicious actions and security policies defining the appropriate responsive actions to rule violations. The system includes an interface for receiving incoming Downloadable and requests made by the Downloadable. The system still further includes a comparator coupled to the interface for examining the Downloadable, requests made by the Downloadable and runtime events to determine whether a security policy has been violated, and a response engine coupled to the comparator for performing a violation-based responsive action.

8 Claims, 6 Drawing Sheets


START
500
RECOGNIZE RECEIPT OF A REQUEST MADE BY A DOWNLOAD ABLE DURING RUNTIME ?
505
NO
YES
INTERRUPT PROCESSING OF THE REQUEST
506
FORWARD A MESSAGE IDENTIFYING THE DOWNLOADABLE TO THE EVENT ROUTER
508
INFORM USER
510
LOG EVENTS
515
ARE ANY INCOMING DOWNLOADABLES KNOWN TO BE SUSPICIOUS ?
520
YES
NO
ARE THE EXECUTING DOWNLOADABLES IN VIOLATION OF A RULE ?
525
NO
540
RESUME OPERATION OF THE DOWNLOAD ABLE
YES
MANAGE THE SUSPICIOUS DOWNLOADABLES
530
535
END ?
NO
YES
END

OTHER PUBLICATIONS

"Java Security: Issues & Solutions" Article published on the Internet by Finjan Software Ltd., 1996, 8 pages.

Mark LaDue, "Online Business Consultant" Article published on the Internet, Home Page, Inc. 1996, 4 pages.

Jim K. Omura, "Novel Applications of Cryptography in Digital Communications", IEEE Communications Magazine, May 1990; pp. 21–27.

Norvin Leach et al, "IE 3.0 Applets Will Earn Certification", PC Week, v13, n29, 2 pages, Jul. 22, 1996.

Microsoft Authenticode Technology, "Ensuring Accountability and Authenticity for Software Components on the Internet", Microsoft Corporation, Oct. 1996, including contents, Introduction and pp. 1–10.

Web page: http://iel.ihs.com:80/cgi–bin/iel_cgi?se . . . 2ehts%26ViewTemplate%3ddocview%5fb%2ehts, Okamato, E. et al., "ID–Based Authentication System For Computer Virus Detection", IEEE/IEE Electronic Library online, Electronics Letters, vol. 26, Issue 15, ISSN 0013–5194, Jul. 19, 1990, Abstract and pp. 1169–1170.

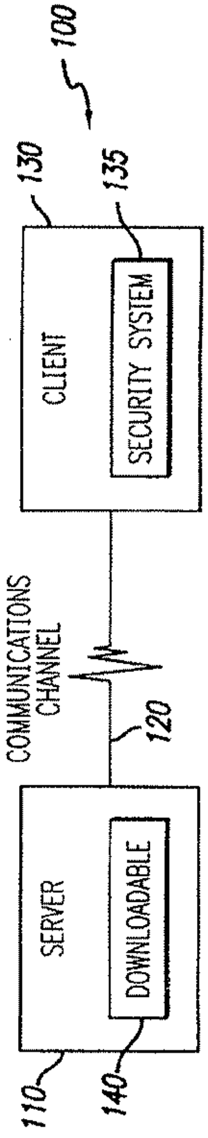

100
130
135
CLIENT
SECURITY SYSTEM
COMMUNICATIONS CHANNEL
120
SERVER
DOWNLOADABLE
110
140

FIG. 1

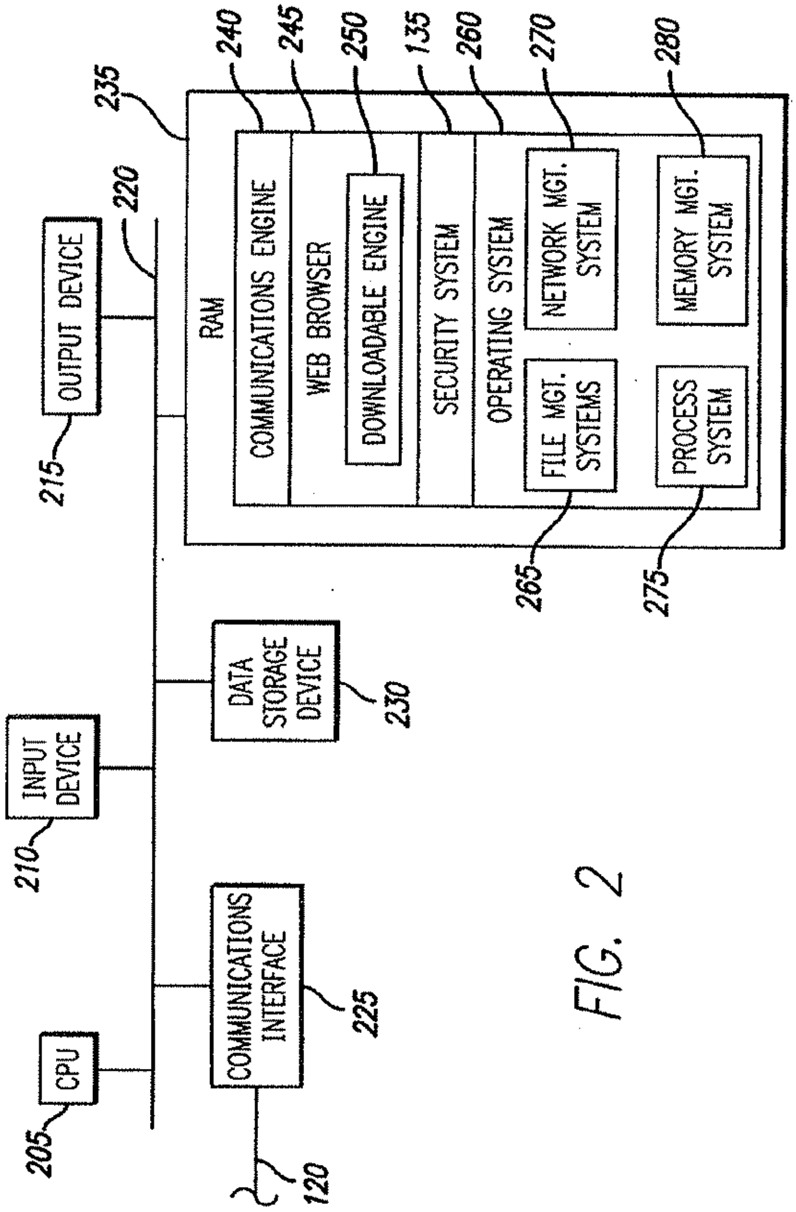
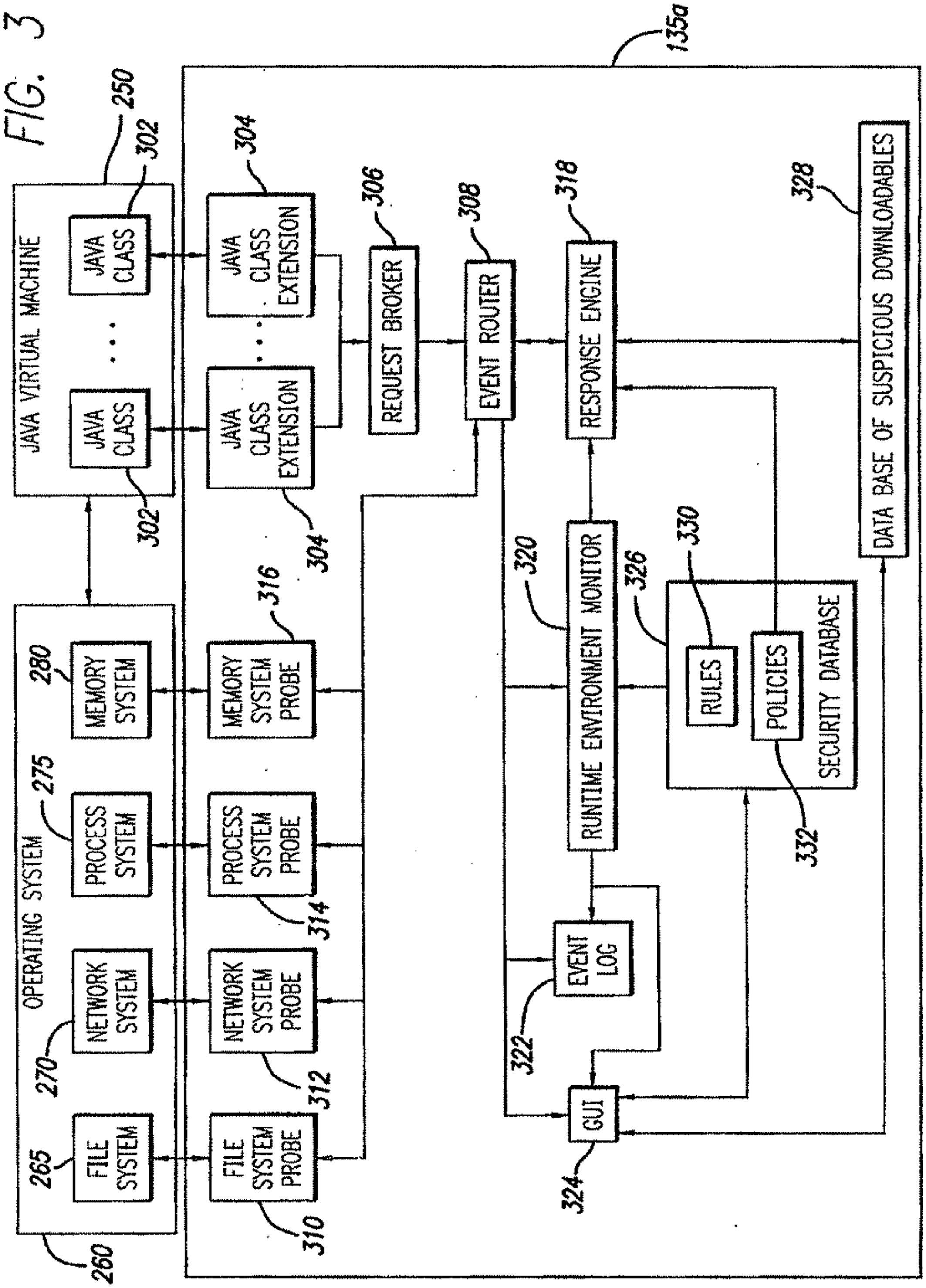

235
240
245
250
135
260
270
280
220
OUTPUT DEVICE
RAM
COMMUNICATIONS ENGINE
WEB BROWSER
DOWNLOADABLE ENGINE
SECURITY SYSTEM
OPERATING SYSTEM
NETWORK MGT. SYSTEM
MEMORY MGT. SYSTEM
FILE MGT. SYSTEMS
PROCESS SYSTEM
215
265
275
DATA STORAGE DEVICE
230
INPUT DEVICE
210
COMMUNICATIONS INTERFACE
225
CPU
205
120

FIG. 2

FIG. 3


135a
250
JAVA VIRTUAL MACHINE
302
JAVA CLASS
. . .
JAVA CLASS
302
260
OPERATING SYSTEM
265
FILE SYSTEM
270
NETWORK SYSTEM
275
PROCESS SYSTEM
280
MEMORY SYSTEM
304
JAVA CLASS EXTENSION
. . .
JAVA CLASS EXTENSION
304
306
REQUEST BROKER
308
EVENT ROUTER
318
RESPONSE ENGINE
328
DATA BASE OF SUSPICIOUS DOWNLOADABLES
316
MEMORY SYSTEM PROBE
314
PROCESS SYSTEM PROBE
312
NETWORK SYSTEM PROBE
310
FILE SYSTEM PROBE
320
RUNTIME ENVIRONMENT MONITOR
326
330
RULES
POLICIES
332
SECURITY DATABASE
322
EVENT LOG
324
GUI

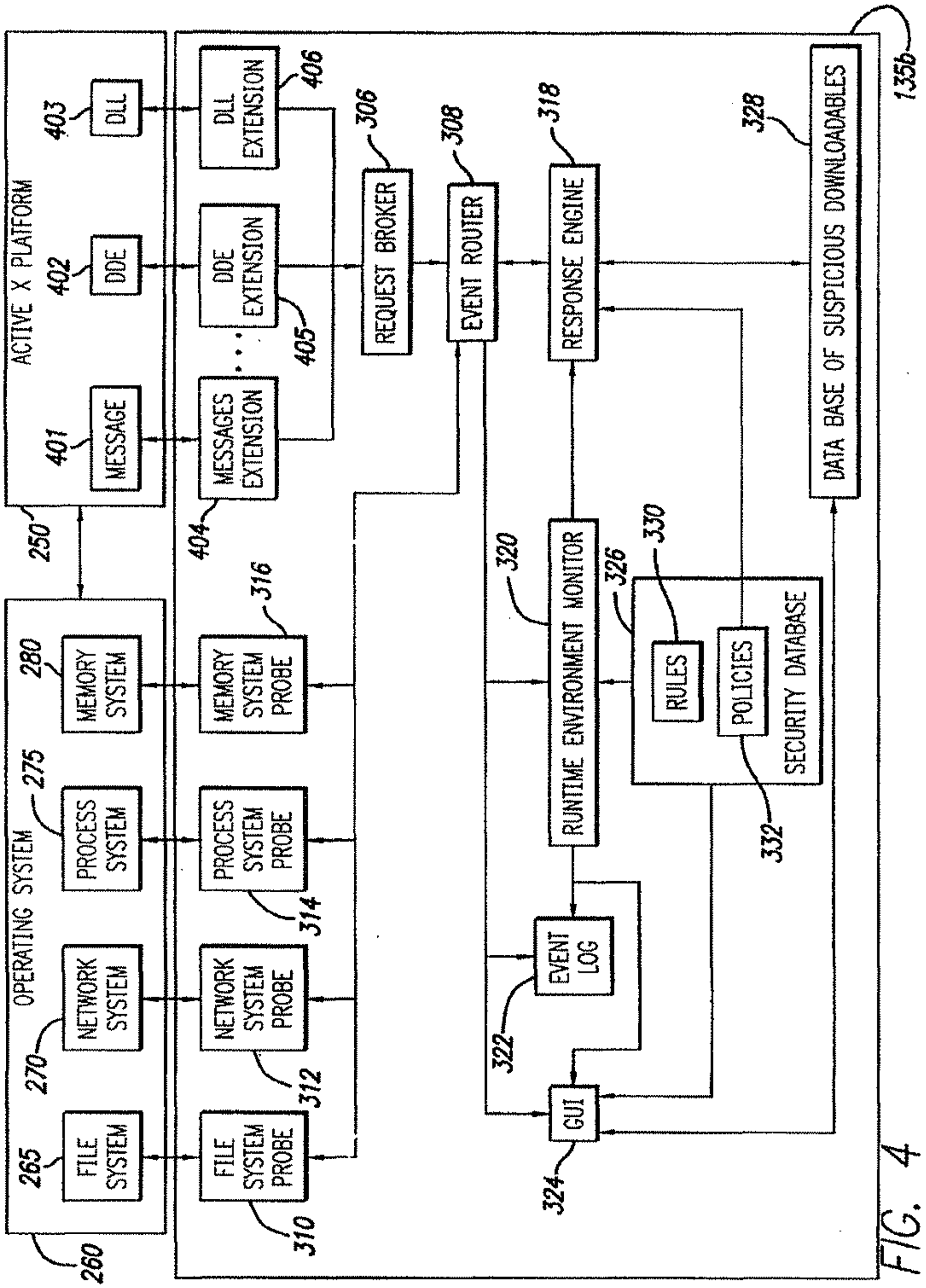

ACTIVE X PLATFORM
403
DLL
402
DDE
401
MESSAGE
250
406
DLL EXTENSION
DDE EXTENSION
405
MESSAGES EXTENSION
404
306
REQUEST BROKER
308
EVENT ROUTER
318
RESPONSE ENGINE
328
DATA BASE OF SUSPICIOUS DOWNLOADABLES
135b
OPERATING SYSTEM
280
MEMORY SYSTEM
275
PROCESS SYSTEM
270
NETWORK SYSTEM
265
FILE SYSTEM
260
316
MEMORY SYSTEM PROBE
PROCESS SYSTEM PROBE
314
NETWORK SYSTEM PROBE
312
FILE SYSTEM PROBE
310
320
RUNTIME ENVIRONMENT MONITOR
326
330
RULES
POLICIES
SECURITY DATABASE
332
EVENT LOG
322
GUI
324

FIG. 4

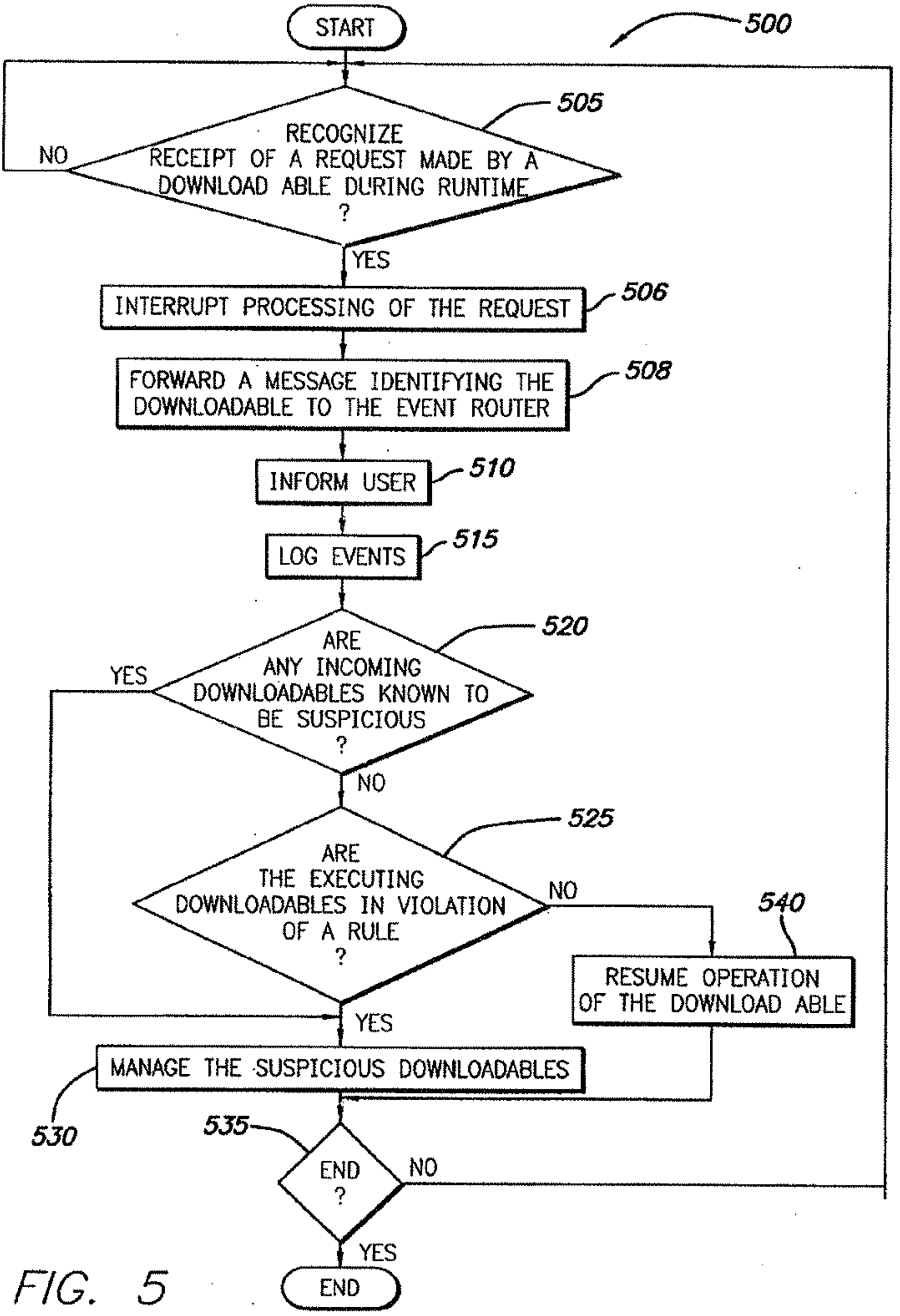

START
500
505
NO
RECOGNIZE RECEIPT OF A REQUEST MADE BY A DOWNLOAD ABLE DURING RUNTIME ?
YES
INTERRUPT PROCESSING OF THE REQUEST
506
FORWARD A MESSAGE IDENTIFYING THE DOWNLOADABLE TO THE EVENT ROUTER
508
INFORM USER
510
LOG EVENTS
515
520
YES
ARE ANY INCOMING DOWNLOADABLES KNOWN TO BE SUSPICIOUS ?
NO
525
ARE THE EXECUTING DOWNLOADABLES IN VIOLATION OF A RULE ?
NO
540
RESUME OPERATION OF THE DOWNLOAD ABLE
YES
MANAGE THE SUSPICIOUS DOWNLOADABLES
530
535
END ?
NO
YES
END

FIG. 5

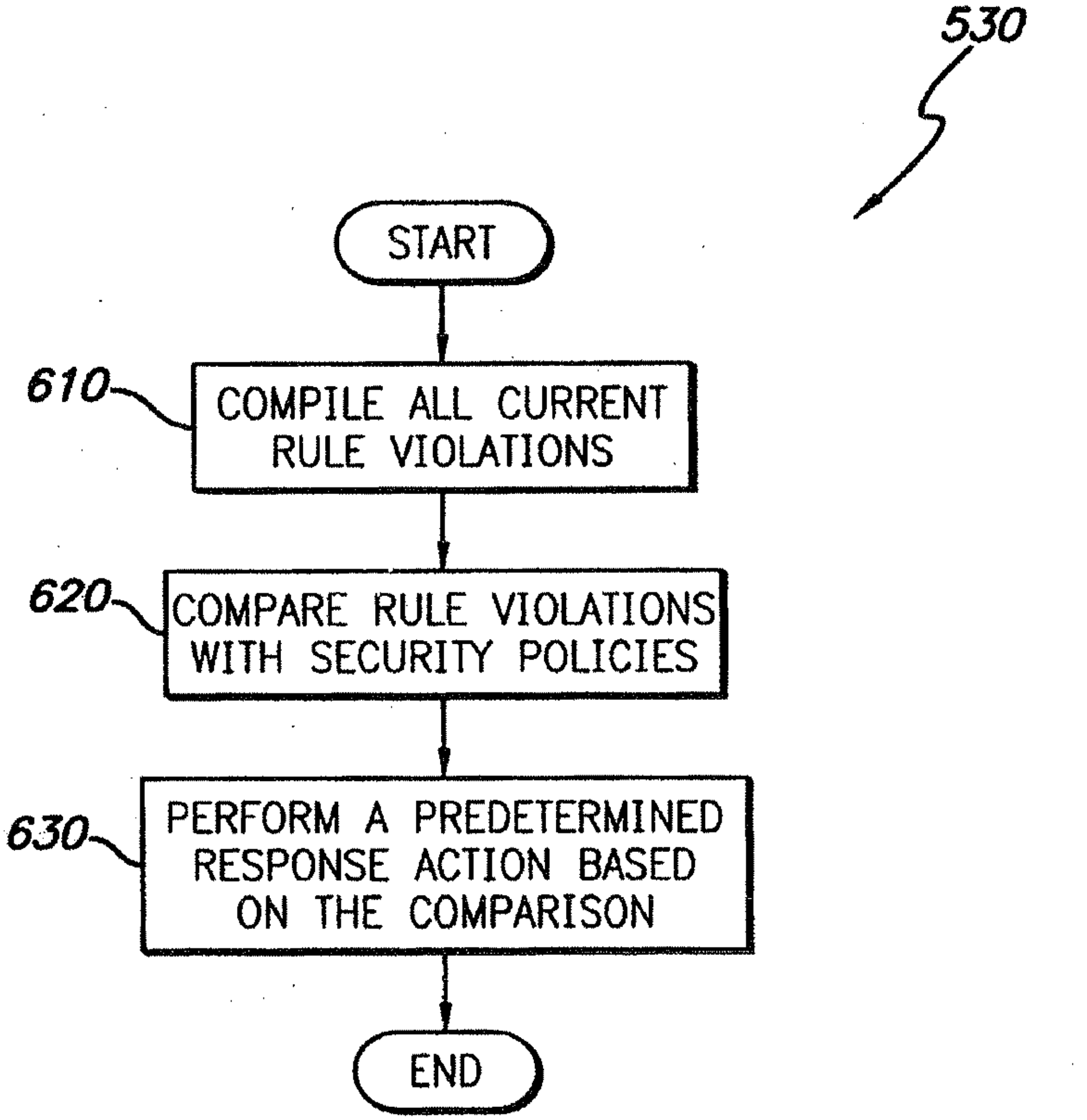

530
START
610
COMPILE ALL CURRENT RULE VIOLATIONS
620
COMPARE RULE VIOLATIONS WITH SECURITY POLICIES
630
PERFORM A PREDETERMINED RESPONSE ACTION BASED ON THE COMPARISON
END

FIG. 6

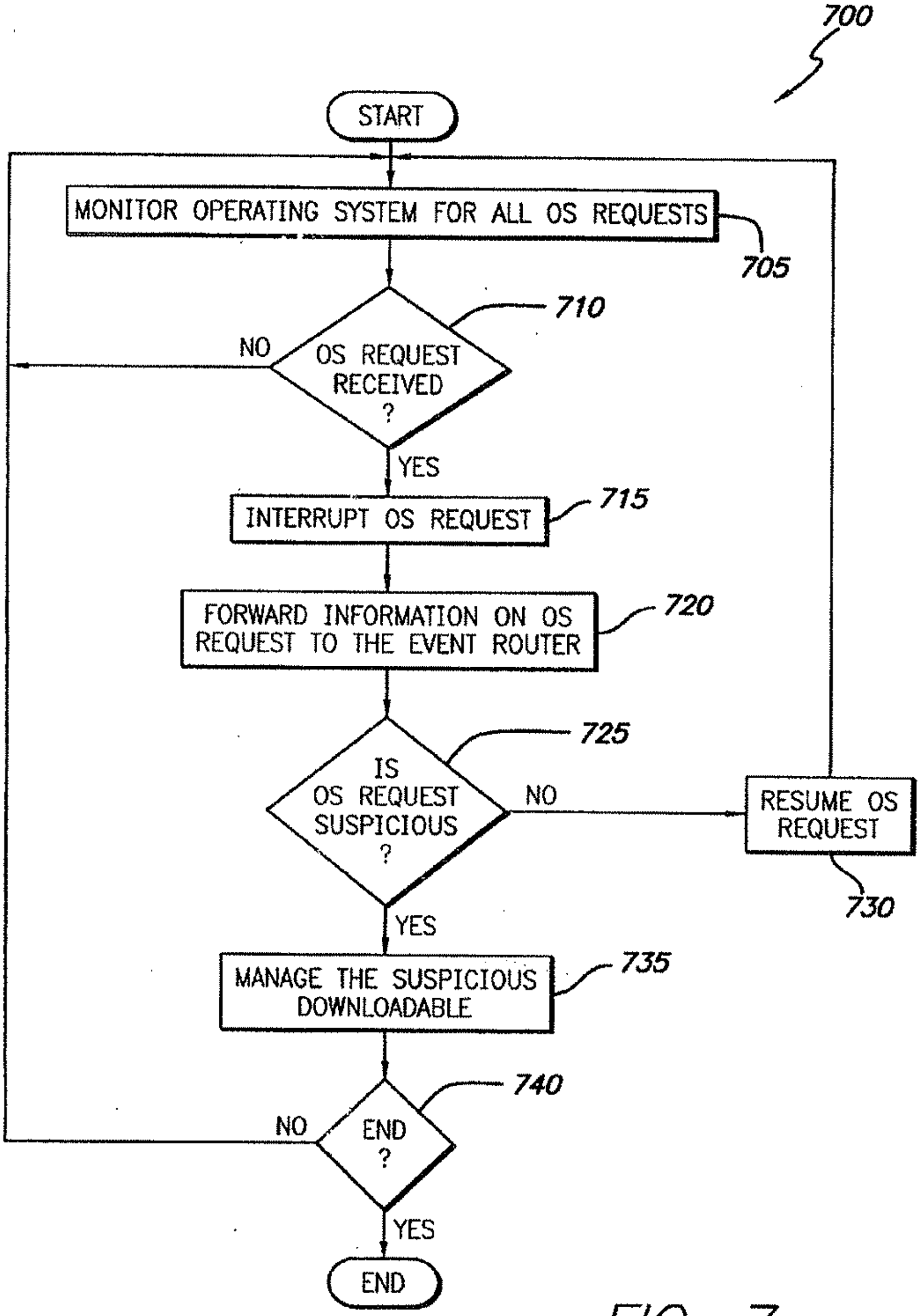

700
START
MONITOR OPERATING SYSTEM FOR ALL OS REQUESTS
705
OS REQUEST RECEIVED ?
710
NO
YES
INTERRUPT OS REQUEST
715
FORWARD INFORMATION ON OS REQUEST TO THE EVENT ROUTER
720
IS OS REQUEST SUSPICIOUS ?
725
NO
RESUME OS REQUEST
730
YES
MANAGE THE SUSPICIOUS DOWNLOADABLE
735
END ?
740
NO
YES
END

FIG. 7

SYSTEM AND METHOD FOR PROTECTING A CLIENT DURING RUNTIME FROM HOSTILE DOWNLOADABLES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is related to co-pending provisional patent application filed on Nov. 8, 1996, entitled "System and Method for Protecting a Computer from Hostile Downloadables," Ser. No. 60/030,639, by inventor Shlomo Touboul, which subject matter is hereby incorporated by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to computer networks, and more particularly to a system and method for protecting clients from hostile Downloadables.

2. Description of the Background Art

The Internet currently interconnects about 100,000 individual computer networks and several million computers. Because it is public, the Internet has become a major source of many system damaging and system fatal application programs, commonly referred to as "viruses."

In response to the widespread generation and distribution of computer viruses, programmers continue to design and update security systems for blocking these viruses from attacking both individual and network computers. On the most part, these security systems have been relatively successful. However, these security systems are typically not configured to recognize computer viruses which have been attached to or masked as harmless Downloadables (i.e., applets). A Downloadable is a small executable or interpretable application program which is downloaded from a source computer and run on a destination computer. A Downloadable is used in a distributed environment such as in the Java™ distributed environment produced by Sun Microsystems or in the ActiveX™ distributed environment produced by Microsoft Corporation.

Hackers have developed hostile Downloadables designed to penetrate security holes in Downloadable interpreters. In response, Sun Microsystems, Inc. has developed a method of restricting Downloadable access to resources (file system resources, operating system resources, etc.) on the destination computer, which effectively limits Downloadable functionality at the Java™ interpreter. Sun Microsystems, Inc. has also provided access control management for basing Downloadable-accessible resources on Downloadable type. However, the above approaches are difficult for the ordinary web surfer to manage, severely limit Java™ performance and functionality, and insufficiently protect the destination computer.

Other security system designers are currently considering digital signature registration stamp techniques, wherein, before a web browser will execute a Downloadable, the Downloadable must possess a digital signature registration stamp. Although a digital signature registration stamp will diminish the threat of Downloadables being intercepted, exchanged or corrupted, this approach only partially addresses the problem. This method does not stop a hostile Downloadable from being stamped with a digital signature, and a digital signature does not guarantee that a Downloadable is harmless. Therefore, a system and method are needed for protecting clients from hostile Downloadables.

SUMMARY OF THE INVENTION

The present invention provides a system for protecting a client from hostile Downloadables. The system includes security rules defining suspicious actions such as WRITE operations to a system configuration file, overuse of system memory, overuse of system processor time, etc. and security policies defining the appropriate responsive actions to rule violations such as terminating the applet, limiting the memory or processor time available to the applet, etc. The system includes an interface, such as Java™ class extensions and operating system probes, for receiving incoming Downloadable and requests made by the Downloadable. The system still further includes a comparator coupled to the interface for examining the Downloadable, requests made by the Downloadable and runtime events to determine whether a security policy has been violated, and a response engine coupled to the comparator for performing the violation-based responsive action.

The present invention further provides a method for protecting a client from hostile Downloadables. The method includes the steps of recognizing a request made by a Downloadable during runtime, interrupting processing of the request, comparing information pertaining to the Downloadable against a predetermined security policy, recording all rule violations in a log, and performing a predetermined responsive action based on the comparison.

It will be appreciated that the system and method of the present invention use at least three hierarchical levels of security. A first level examines the incoming Downloadables against known suspicious Downloadables. A second level examines runtime events. A third level examines the Downloadables operating system requests against predetermined suspicious actions. Thus, the system and method of the invention are better able to locate hostile operations before client resources are damaged.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating a network system in accordance with the present invention;

FIG. 2 is a block diagram illustrating details of the client;

FIG. 3 is a block diagram illustrating details of a security system;

FIG. 4 is a block diagram illustrating details of an alternative security system;

FIG. 5 is a flowchart illustrating a method for protecting a client from suspicious Downloadables;

FIG. 6 is a flowchart illustrating the method for managing a suspicious Downloadable; and

FIG. 7 is a flowchart illustrating a supplementary method for protecting a client from suspicious Downloadables.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 is a block diagram illustrating a network system **100** in accordance with the present invention. Network system **100** includes a server **110** coupled to a communications channel **120**, e.g., an Internet or an Intranet. The communications channel **120** is in turn coupled to a client **130**, e.g., an individual computer, a network computer, a kiosk workstation, etc., which includes a security system **135** for protecting the client **130** from hostile (i.e., will adversely effect the operational characteristics of the client **130**) or suspicious (i.e., potentially hostile) downloadables.

Server **110** forwards a Downloadable **140** across the communications channel **120** to the client **130**. During runtime, the security system **135** examines each Downloadable **140** and the actions of each Downloadable **140** to monitor for hostile or suspicious actions.

FIG. 2 is a block diagram illustrating details of a client **130**, which includes a Central Processing Unit (CPU) **205**, such as a Motorola Power PC® microprocessor or an Intel Pentium® microprocessor, coupled to a signal bus **220**. The client **130** further includes an input device **210** such as a keyboard and mouse, an output device **215** such as a Cathode Ray Tube (CRT) display, a data storage device **230** such as Read Only Memory (ROM) or magnetic disk, and a Random-Access Memory (RAM) **235**, each being coupled to signal bus **220**. A communications interface **225** is coupled between the communications channel **120** and the signal bus **220**.

An operating system **260** controls processing by CPU **205**, and is typically stored in data storage device **230** and loaded into RAM **235** for execution. The operating system **260** includes a file management system **265**, a network management system **270**, a process system **275** for controlling CPU **205**, and a memory management system **280** for controlling memory use and allocation. A communications engine **240** generates and transfers message packets to and from the communications channel **140** via the communications interface **225**, and may also be stored in data storage device **230** and loaded into RAM **235** for execution.

The client **130** further includes a web browser **245**, such as the Netscape™ web browser produced by the Netscape Corporation, the Internet Explorer™ web browser produced by the Microsoft Corporation, or the Java™ Developers Kit 1.0 web browser produced by Sun Microsystems, Inc., for communicating via the communications channel **120**. The web browser **245** includes a Downloadable engine **250** for managing and executing received Downloadables **140**.

The client **130** further includes the security system **135** as described with reference to FIG. **1**. The security system **135** may be stored in data storage device **230** and loaded into RAM **235** for execution. During runtime, the security system **135** intercepts and examines Downloadables **140** and the actions of Downloadables **140** to monitor for hostile or suspicious actions. If the security system **135** recognizes a suspicious Downloadable **140** or a suspicious request, then the security system **135** can perform an appropriate responsive action such as terminating execution of the Downloadable **140**.

FIG. **3** is a block diagram illustrating details of the security system **135***a*, which is a first embodiment of security system **135** of FIG. **2** when operating in conjunction with a Java™ virtual machine **250** (i.e., the Downloadable engine **250**) that includes conventional Java™ classes **302**. Each of the Java™ classes **302** performs a particular service such as loading applets, managing the network, managing file access, etc. Although Downloadables are being described with reference to the Java™ distributed environment, Downloadables herein correspond to all downloadable executable or interpretable programs for use in any distributed environment such as in the ActiveX™ distributed environment.

Examples of Java™ classes used in Netscape Navigator™ include AppletSecurity.class, EmbeddedAppletFrame.class, AppletClassLoader.class, MozillaAppletContext.class, ServerSocket.class, SecurityException.class and SecurityManager.class, etc. Examples of Java™ classes used in Internet Explorer™ include AppletSecurity.class, BrowserAppletFrame.class, AppletClassLoader.class, ServerSocket.class, SecurityException.class and SecurityManager.class, etc. Other classes may include Broker.class, BCInterface.class, SocketConnection.class, queueManager.class, BrowserExtension.class, Message.class, MemoryMeter.class and AppletDescription.class.

The security system **135***a* includes Java™ class extensions **304**, wherein each extension **304** manages a respective one of the Java™ classes **302**. When a new applet requests the service of a Java class **302**, the corresponding Java™ class extension **304** interrupts the request and generates a message to notify the request broker **306** of the Downloadable's request. The request broker **306** uses TCP/IP message passing protocol to forward the message to the event router **308**.

The security system **135***a* further includes operating system probes **310**, **312**, **314** and **316**. More particularly, a file management system probe **310** recognizes applet instructions sent to the file system **265** of operating system **260**, a network system probe **312** recognizes applet instructions sent to the network management system **270** of operating system **260**, a process system probe **314** recognizes applet instructions sent to the process system **275** of operating system **260**, and a memory management system probe **316** recognizes applet instructions sent to the memory system **280** of operating system **260**. When any of the probes **310–316** recognizes an applet instruction, the recognizing probe **310–316** sends a message to inform the event router **308**.

Upon receipt of a message, the event router **308** accordingly forwards the message to a Graphical User Interface (GUI) **324** for notifying the user of the request, to an event log **322** for recording the message for subsequent analysis, and to a runtime environment monitor **320** for determining whether the request violates a security rule **330** stored in a security database **326**. Security rules **330** include a list of computer operations which are deemed suspicious. Suspicious operations may include READ/WRITE operations to a system configuration file, READ/WRITE operations to a document containing trade secrets, overuse of system memory, overuse of system processor time, too many applets running concurrently, or too many images being displayed concurrently. For example, the runtime environment monitor **320** may determine that a security rule **330** has been violated when it determines that an applet uses more than two megabytes of RAM **235** or when the Java™ virtual machine **250** runs more than five applets concurrently.

Upon recognition of a security rule **330** violation, the runtime environment monitor **320** records the violation with the event log **322**, informs the user of the violation via the GUI **324** and forwards a message to inform the response engine **318** of the violation. The response engine **318** analyzes security policies **332** stored in the security database **326** to determine the appropriate responsive action to the rule **330** violation. Appropriate responsive actions may include terminating the applet, limiting the memory or processor time available to the applet, etc. For example, the response engine **318** may determine that a security policy **332** dictates that when more than five applets are executed concurrently, operation of the applet using the greatest amount of RAM **235** should be terminated. Further, a security policy **332** may dictate that when an applet or a combination of applets violates a security policy **332**, the response engine **318** must add information pertaining to the applet or applets to the suspicious Downloadables database **328**. Thus, when the applet or applets are encountered again, the response engine **318** can stop them earlier.

The GUI **324** enables a user to add or modify the rules **330** of the security database **326**, the policies **332** of the security database **326** and the suspicious applets of the suspicious Downloadables database **328**. For example, a user can use the GUI **324** to add to the suspicious Downloadables database **328** applets generally known to be hostile, applets deemed to be hostile by the other clients **130** (not shown), applets deemed to be hostile by network MIS managers, etc. Further, a user can use the GUI **324** to add to the rules **330** actions generally known to be hostile, actions deemed to be hostile by network MIS managers, etc.

It will be appreciated that the embodiment illustrated in FIG. **3** includes three levels of security. The first level examines the incoming Downloadables **140** against known suspicious Downloadables. The second level examines the Downloadables' access to the Java™ classes **302**. The third level examines the Downloadables requests to the operating system **260**. Thus, the security system **135***a* is better apt to locate a hostile operation before an operation damages client **130** resources.

FIG. **4** is a block diagram illustrating details of a security system **135***b*, which is a second embodiment of security system **135** when operating in conjunction with the ActiveX™ platform (i.e., the Downloadable engine **250**) which uses message **401** calls, Dynamic-Data-Exchange (DDE) **402** calls and Dynamically-Linked-Library (DLL) **403** calls. Thus, instead of having Java™ class extensions **304**, the security system **135** has a messages extension **401** for recognizing message **401** calls, a DDE extension **405** for recognizing DDE **402** calls and a DLL extension **406** for recognizing DLL calls. Upon recognition of a call, each of the messages extension **404**, the DDE extension **405** and the DLL extension **406** send a message to inform the request broker **306**. The request broker **306** and the remaining elements operate similarly to the elements described with reference to FIG. **3**.

FIG. **5** is a flowchart illustrating a method **500** for protecting a client **130** from hostile and suspicious Downloadables **140**. Method **500** begins with the extensions **304**, **404**, **405** or **406** in step **505** waiting to recognize the receipt of a request made by a Downloadable **140**. Upon recognition of a request, the recognizing extension **304**, **404**, **405** or **406** in step **506** interrupts processing of the request and in step **508** generates and forwards a message identifying the incoming Downloadable **140** to the request broker **306**, which forwards the message to the event router **308**.

The event router **308** in step **510** forwards the message to the GUI **324** for informing the user and in step **515** to the event log **322** for recording the event. Further, the event router **308** in step **520** determines whether any of the incoming Downloadables **140** either alone or in combination are known or previously determined to be suspicious. If so, then method **500** jumps to step **530**. Otherwise, the runtime environment monitor **320** and the response engine **318** in step **525** determine whether any of the executing Downloadables **140** either alone or in combination violate a security rule **330** stored in the security database **332**.

If a rule **330** has been violated, then the response engine **318** in step **530** manages the suspicious Downloadable **140**. Step **530** is described in greater detail with reference to FIG. **6**. Otherwise, if a policy has not been violated, then response engine **318** in step **540** resumes operation of the Downloadable **140**. In step **535**, a determination is made whether to end method **500**. For example, if the user disconnects the client **130** from the server **110**, method **500** ends. If a request to end is made, then method **500** ends. Otherwise, method **500** returns to step **505**.

FIG. **6** is a flowchart illustrating details of step **530**. Since multiple rule **330** violations may amount to a more serious violation and thus require a stricter response by the response engine **318**, step **530** begins with the response engine **318** in step **610** compiling all rule **330** violations currently occurring. The response engine **318** in step **620** compares the compiled rule **330** violations with the security policies **332** to determine the appropriate responsive action for managing the suspicious Downloadable **140** or Downloadables **140**, and in step **630** the response engine **318** performs a predetermined responsive action. Predetermined responsive actions may include sending a message via the GUI **324** to inform the user, recording the message in the event log **322**, stopping execution of a suspicious Downloadable **140**, storing a Downloadable **140** or combination of Downloadables **140** in the suspicious Downloadable database **328**, limiting memory available to the Downloadable **140**, limiting processor time available to the Downloadable **140**, etc.

FIG. **7** is a flowchart illustrating a supplementary method **700** for protecting a client **130** from suspicious Downloadables **140**. Method **700** begins with operating system probes **310**, **312**, **314** and **316** in step **705** monitoring the operating system **260** for Operating System (OS) requests from Downloadables **140**. As illustrated by step **710**, when one of the probes **310**–**316** recognizes receipt of an OS request, the recognizing probe **310**–**316** in step **715** interrupts the request and in step **720** forwards a message to inform the event router **308**.

The event router **308** in step **725** routes the information to each of the components of the security engine **135** as described with reference to FIG. **5**. That is, the event router **308** forwards the information to the GUI **324** for informing the user, to the event log **322** for recordation and to the runtime environment monitor **320** for determining if the OS request violates a rule **330**. The response engine **318** compares the OS request alone or in combination with other violations against security policies **332** to determine the appropriate responsive actions. It will be appreciated that, based on the security policies **332**, the response engine **318** may determine that an OS request violation in combination with other OS request violations, in combination with rule **330** violations, or in combination with both other OS request violations and rule **330** violations merits a stricter responsive action.

If the OS request does not violate a security rule **330**, then the response engine **318** in step **730** instructs the operating system **260** via the recognizing probe **310**–**316** to resume operation of the OS request. Otherwise, if the OS request violates a security rule **330**, then the response engine **318** in step **730** manages the suspicious Downloadable by performing the appropriate predetermined responsive actions as described with reference to FIGS. **5** and **6**. In step **740**, a determination is made whether to end method **700**. If a request to end the method is made, then method **700** ends. Otherwise, method **700** returns to step **705**.

The foregoing description of the preferred embodiments of the invention is by way of example only, and other variations of the above-described embodiments and methods are provided by the present invention. For example, although the invention has been described in a system for protecting an internal computer network, the invention can be embodied in a system for protecting an individual computer. Components of this invention may be implemented using a programmed general purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. The embodiments described herein have been presented for purposes of illustration and are not intended to be exhaustive or limiting. Many variations and modifications are possible in light of the foregoing teaching. The system is limited only by the following claims.

What is claimed is:

1. A computer-based method, comprising:

monitoring the operating system during runtime for an event caused from a request made by a Downloadable;

interrupting processing of the request;

comparing information pertaining to the Downloadable against a predetermined security policy; and

performing a predetermined responsive action based on the comparison, the predetermined responsive action including storing results of the comparison in an event log.

2. A computer-based method, comprising:

monitoring the operating system during runtime for an event caused from a request made by a Downloadable;

interrupting processing of the request;

comparing information pertaining to the Downloadable against a predetermined security policy; and

performing a predetermined responsive action based on the comparison, the predetermined responsive action including storing the Downloadable in a suspicious Downloadable database.

3. A system, comprising:

a security policy;

an operating system interface for recognizing a runtime event caused from a request made by a Downloadable;

a comparator coupled to the interface for comparing information pertaining to the received Downloadable with the security policy;

a response engine coupled to the comparator for performing a predetermined responsive action based on the comparison with the security policy; and

an event log coupled to the comparator for storing results of the comparison.

4. A system, comprising:

a security policy;

an operating system interface for recognizing a runtime event caused from a request made by a Downloadable;

a comparator coupled to the interface for comparing information pertaining to the received Downloadable with the security policy;

a response engine coupled to the comparator for performing a predetermined responsive action based on the comparison with the security policy; and

a suspicious Downloadable database for storing known and previously-deemed suspicious Downloadables.

5. A system for determining whether a Downloadable, which is received by a Downloadable engine, is suspicious, comprising:

means for monitoring the operating system during runtime for an event caused from a request made by a Downloadable;

means for interrupting processing of the request;

means for comparing information pertaining to the Downloadable against a predetermined security policy; and

means for performing a predetermined responsive action based on the comparison, the predetermined responsive action including storing results of the comparison in an event log.

6. A system for determining whether a Downloadable, which is received by a Downloadable engine, is suspicious, comprising:

means for monitoring the operating system during runtime for an event caused from a request made by a Downloadable;

means for interrupting processing of the request;

means for comparing information pertaining to the Downloadable against a predetermined security policy; and

means for performing a predetermined responsive action based on the comparison, the predetermined responsive action including storing the Downloadable in a suspicious Downloadable database.

7. A computer-readable storage medium storing program code for causing a computer to perform the steps of:

monitoring the operating system during runtime for an event caused from a request made by a Downloadable;

interrupting processing of the request;

comparing information pertaining to the Downloadable against a predetermined security policy; and

performing a predetermined responsive action based on the comparison, the predetermined responsive action including storing results of the comparison in an event log.

8. A computer-readable storage medium storing program code for causing a computer to perform the steps of:

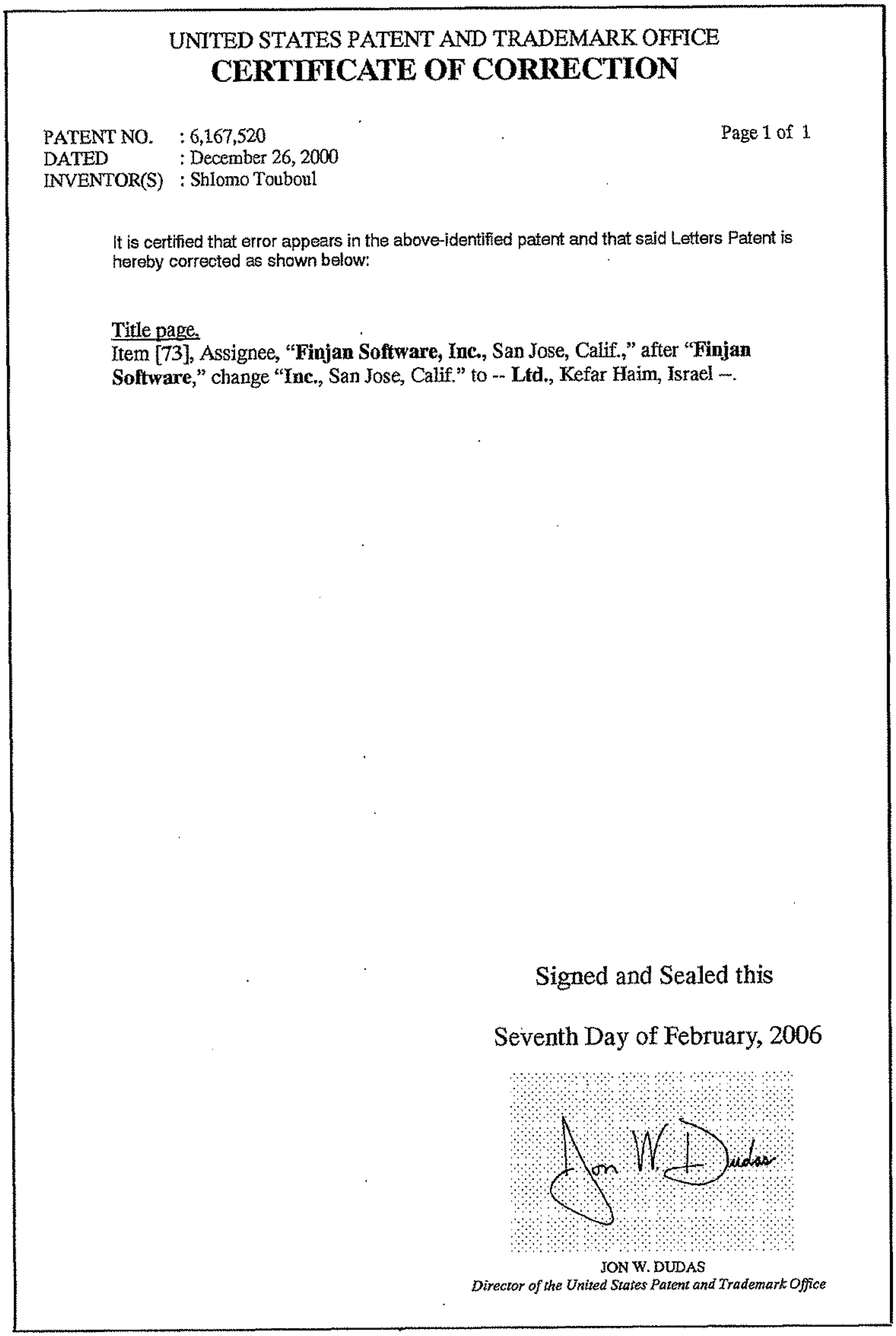
monitoring the operating system during runtime for an event caused from a request made by a Downloadable;

interrupting processing of the request;

comparing information pertaining to the Downloadable against a predetermined security policy; and

performing a predetermined responsive action based on the comparison, the predetermined responsive action including storing the Downloadable in a suspicious Downloadable database.

* * * * *



UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO.	: 6,167,520
DATED	: December 26, 2000
INVENTOR(S)	: Shlomo Touboul

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Title page,
Item [73], Assignee, "**Finjan Software, Inc.**, San Jose, Calif.," after "**Finjan Software**," change "**Inc.**, San Jose, Calif." to -- **Ltd.**, Kefar Haim, Israel --.

Signed and Sealed this

Seventh Day of February, 2006

Jon W. Dudas

JON W. DUDAS
Director of the United States Patent and Trademark Office

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/495,157	02/27/2007	7185361	NAI1P075/99.039.01	4471

21186 7590 02/07/2007

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. BOX 2938
MINNEAPOLIS, MN 55402

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Extension under 35 U.S.C. 154 (b)
(application filed after June 7, 1995 but prior to May 29, 2000)

The Patent Term Extension is 0 day(s). Any patent to issue from the above-identified application will include an indication of the 0 day extension on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Extension is the filing date of the most recent CPA.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

Thomas D. Ashoff, Mt Airy, MD;
Steve O. Chew, Pittsburgh, PA;
Jeffrey J. Graham, Olney, MD;
Andrew J. Mullican, Columbia, MD;

IR103 (Rev. 11/05)

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/495,157	01/31/2000	Thomas D. Ashoff	NAI1P075/99.039.01	4471

7590 09/10/2003

Schwegman, Lundberg, Woessner &
Kluth, P.A.
P.O. Box 2938
Minneapolis, MN 55402

EXAMINER	
MASHAAL, ALI M	
ART UNIT	**PAPER NUMBER**
2133	10

DATE MAILED: 09/10/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 07-01)

Office Action Summary	**Application No.** 09/495,157	**Applicant(s)** ASHOFF ET AL.
	Examiner Ali M. Mashaal	**Art Unit** 2133

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- *Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).*
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 January 2000* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-17* is/are pending in the application.

4a) Of the above claim(s) ______ is/are withdrawn from consideration.

5)☐ Claim(s) ______ is/are allowed.

6)☒ Claim(s) *1-17* is/are rejected.

7)☐ Claim(s) ______ is/are objected to.

8)☐ Claim(s) ______ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *31 January 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on ______ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☒ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. ______ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)
2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3)☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 3 .
4)☐ Interview Summary (PTO-413) Paper No(s). ______ .
5)☐ Notice of Informal Patent Application (PTO-152)
6)☐ Other:

DETAILED ACTION

Oath/Declaration

1. The oath or declaration is defective. A new oath or declaration in compliance with 37 CFR 1.67(a) identifying this application by the application number and filing date is required. See MPEP §§ 602.01 and 602.02.

The oath or declaration is defective because:
The "[ ] is attached to option is marked, when in fact the declaration was filed after the application, and therefore could not have been attached.

Specification

2. The disclosure is objected to because it contains an embedded hyperlink and/or other form of browser-executable code. For example on page 4, lines 11 and 12, "http://www.stanford.edu/~hodges/talks/mactivity.ldap.97/index2.html". Applicant is required to delete **all** embedded hyperlinks and/or other forms of browser-executable code. See MPEP § 608.01.

Drawings

3. Figures 1, 4, and 5 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-7 are rejected as being directed to non-statutory subject matter.

Claim 1 is directed to a system for authorizing client access. However, the system comprises a directory, firewall, and authorization filter. As mentioned on page 2 of the Specification, a firewall can be software alone, see lines 10-12. The directory, asserted in figure 4, is a collection of data. The filter is the criteria used for authentication and is intangible; see page 11, lines 7-10, and page 14, lines 15 and 16. Each of the components in claim 1 is disclosed as being implemented as software alone with no tangible elements.

Claims 2-7 each further limit claim 1 by adding an intangible feature as disclosed below.

Claim 2 limits claim 1, only to specify an LDAP directory and therefore is rejected over claim 1.

Claim 3 limits claim 1, only to specify the filter using a GUI and therefore is rejected over claim 1.

Claims 4 and 5 limit claim 1, only to specify the implementation of a per-user and per-service scheme respectively, and therefore are rejected over claim 1.

Claim 6 limits claim 1, only to specify SSL as the protocol for communication between the firewall and the directory, and therefore is rejected over claim 1.

Claim 7 limits claim 1, only by specifying multiple directories and therefore is rejected over claim 1.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over US patent number 6,131,120 to Reid in view of The Microsoft Computer Dictionary, 1997, further in view of Check Point Account Management Client, Version 1.0, September 1998.

7. In reference to claims 1, 8, and 17, Reid substantially discloses a system comprising at least one directory, column 5, lines 23-31, that can be accessed using a network protocol. Although not explicitly mentioned, the only possible way for the directory to communicate with the various components on the network illustrated in figure 4, is through the use of a network protocol.

Column 5 lines 23-35 and column 6 lines 54-65, discuss the possibility of deleting or omitting a firewall and implementation of the firewall functions into other network elements. However, the examiner first notes that per the Microsoft Computer Dictionary, a firewall is defined as a security system intended to protect an organization's network against external threats; a firewall prevents computers in the organization's network from communicating directly with computers external to the network and vice versa. In Reid, the firewall functions are integrated into a router/gateway receiving information (Router Access List) from a directory server, see column 6 lines 17-26 and lines 57-61. Acting as the firewall, the routers/gateways download the access list, and grant or deny access correspondingly. The examiner asserts that while the reference teaches elimination of a separate firewall, it does not in fact eliminate the firewall altogether but instead combines the functions of a firewall with that of a router/gateway into a single unit.

This firewall is configured to intercept network resource requests, see Reid, column 8, lines 6-11, which outlines that the users are either allowed or denied access by the router/gateway. This means that the router/gateway intercepts the users network requests.

Also, column 8, line 9, says "each user" in reference to users accessing the WAN. This establishes a plurality of users.

As per comparison of the authorization filter to directory entries, the examiner asserts that normal and well-known function of a firewall is to selectively control what client user has access to resources it protects. Thus, some type of authorization filter (i.e. filter relative to authorization information) would have to be executed. In Reid, see column 8 lines 14-21, RAL, or router/gateway access list is sent to each router/gateway and controls who may have access through router/gateway. The examiner asserts that in order for the directory to generate the appropriate access list, each router/gateway must have transmitted its access criteria to the directory. The examiner further asserts that this criteria is an authorization filter and that in order for the directory to send back a correct access list, some comparison must have been made with directory entries and the router/gateway criteria (authorization filter).

As per the filter being generated based on schema that is predefined by the entity, Reid discloses that the access list is all ready set in the master directory and is then downloaded to the router/gateway, column 8, lines 14-21. This asserts that the schema is predefined, and that the filter is generated based on it.

We have established to this point that Reid teaches a system for authorizing client access to a network resource having one or more directories that can be accessed through a network protocol, and a firewall that is configured to intercept network resource requests from a plurality of clients, in which the firewall authorizes the requests of the clients based on one or more entries in the said directory to an authorization filter, wherein the authorization filter is generated based on a directory schema that is predefined by the said entity.

As per the directory being configured to store information concerning an organization's entity, Reid's directory, is configured to store names, workstations, router/gateways, servers, IP addresses, locations and so on, column 5, lines 36-39. Reid's directory does not store information concerning an entity's organization. Check Point Account Management Client (disclosed in applicants IDS) teaches storing an entity's organization in a directory tree, page 2, figure 1-1 LDAP Tree Example. Since this structure can be used to authenticate users, it would have been obvious to one of ordinary skill in the art at the time of the invention to take the analogous storage directory tree of Reid and modify it such that Reid's directory tree stored an entity's organization similar to that if Checkpoint. Examiner also notes that Reid implies that it is not necessarily true that what he chooses to store in the directory is the only thing that can be stored, refer to column 7 line 61, when Reid says "In the embodiment of this invention, the objects may be individual's names....".

8. In reference to claims 2 and 9 which further limit claims 1 and 8 respectively by specifying the directory as being an LDAP directory, Reid teaches a system analogous to that in claims 1 and 8 as mentioned above, and also teaches the use of LDAP directories. See column 4, lines 7-11, column 6, lines 20-24, and column 8, lines 28-31.

9. In reference to claims 4 and 5, which each depend from claim 1, Reid substantially teaches a system analogous to that in claim 1 as mentioned above, and also teaches that the directory contains objects with associated attributes. Specifically, Reid says that the users, router/gateways, and servers are objects. The examiner asserts that Reid's invention teaches both per-user and per-server authentication since this object-oriented directory is organized such that users, router/gateways, and servers are all objects each of which having attributes that include IP address, password, privileges, and location. See column 6, lines 13-20. Accordingly, Reid substantially suggests that any of the two authentication methods could be used.

10. Claims 6 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Reid as applied to claims 1 and 8 above, and further in view of US patent number 5,657,390 to Elgamal. Reid's invention fails to teach the use of SSL as means to secure the communication between the firewall and the directory. Elgamal's analogous

invention teaches that because the socket layer is widely used in networks, integration of SSL into machines that are connected to the network and receive requests would be facilitated. See column 1, lines 60-63. It would have been obvious to one having ordinary skill in the art at the time the invention was made to take Reid's firewall and use the SSL method taught by Elgamal to communicate between the firewall and the directory. One having ordinary skill in the art at the time the invention was made would have been motivated to do so because Elgamal establishes a need for SSL as a security mechanism between various applications to transfer various data between one another. See column 1, lines 44-54. Furthermore, the nature of the information being transferred between the firewail and the directory in the applicant's invention is private and sensitive.

11. In reference to claims 7 and 14, which further limit claims 1 and 8 respectively by specifying multiple directories being queried, Reid substantially teaches a system analogous to that in claims 1 and 8 as mentioned above, and also teaches the use of multiple directories as described in column 7, lines 58-61, when he refers to distributed directories and a master directory.

12. In reference to claims 15 and 16 which further limit claim 8 by specifying that the request comes from an interna! user and an external user respectively, Reid substantially teaches a system analogous to that in claim 8 as mentioned above, and also states that his system handles both internal and external requests. Refer to

column 7, lines 39-44, when Reid says that the security policy is defined whether the user is internal or external to the network.

13. Claims 3 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Reid as applied to claims 1 and 8 above, and further in view of US patent number 5,898,830 to Wesinger. Reid discloses a system that encompasses all the limitations of claims 1 and 8, but fails to teach the use of a GUI interface to specify the authorization filter. Wesinger, in an analogous art, teaches the use of a graphical user point and click web interface for configuring the firewall and specifying configuration parameters. It would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the firewall of Reid to include a GUI interface by which the authorization filter could be specified. One having ordinary skill in the art at the time the invention was made would have been motivated to do so because graphical user interfaces have become highly popular and favored for their ease of use, and because they are cheap and easy to develop.

14. Claims 11 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Reid in view of Wesinger as applied to claims 3 and 10 above, and further in view of Reid as applied to claims 4 and 5 above.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

The following patents are cited to further show the state of the art with respect to directory-based access controlled networks:

U.S. Pat. No. 006047322A to Vaid

Pub. No. 20030126468 to Markham

U.S. Pat. No. 006212558B1 to Antur

U.S. Pat. No. 006233688B1 to Montenegro

U.S. Pat. No. 006324648B1 to Grantges

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ali M. Mashaal whose telephone number is 703-305-7854. The examiner can normally be reached on 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Albert Decady can be reached on 703-305-9595. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-3800.

A M

ALBERT DECADY
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

09/04/03

S/N 09/495157 **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:		Examiner: Ali M Mashaal
Serial No.:	09/495157	Group Art Unit: ~~2133~~ 2136
Filed:	January 31, 2000	Docket No.: 105.201US1
Title:	System, Method and Computer Program Product for Authenticating Users Using a Lightweight Directory Access Protocol (LDAP) Directory Server	

AMENDMENT AND RESPONSE UNDER 37 CFR § 1.111

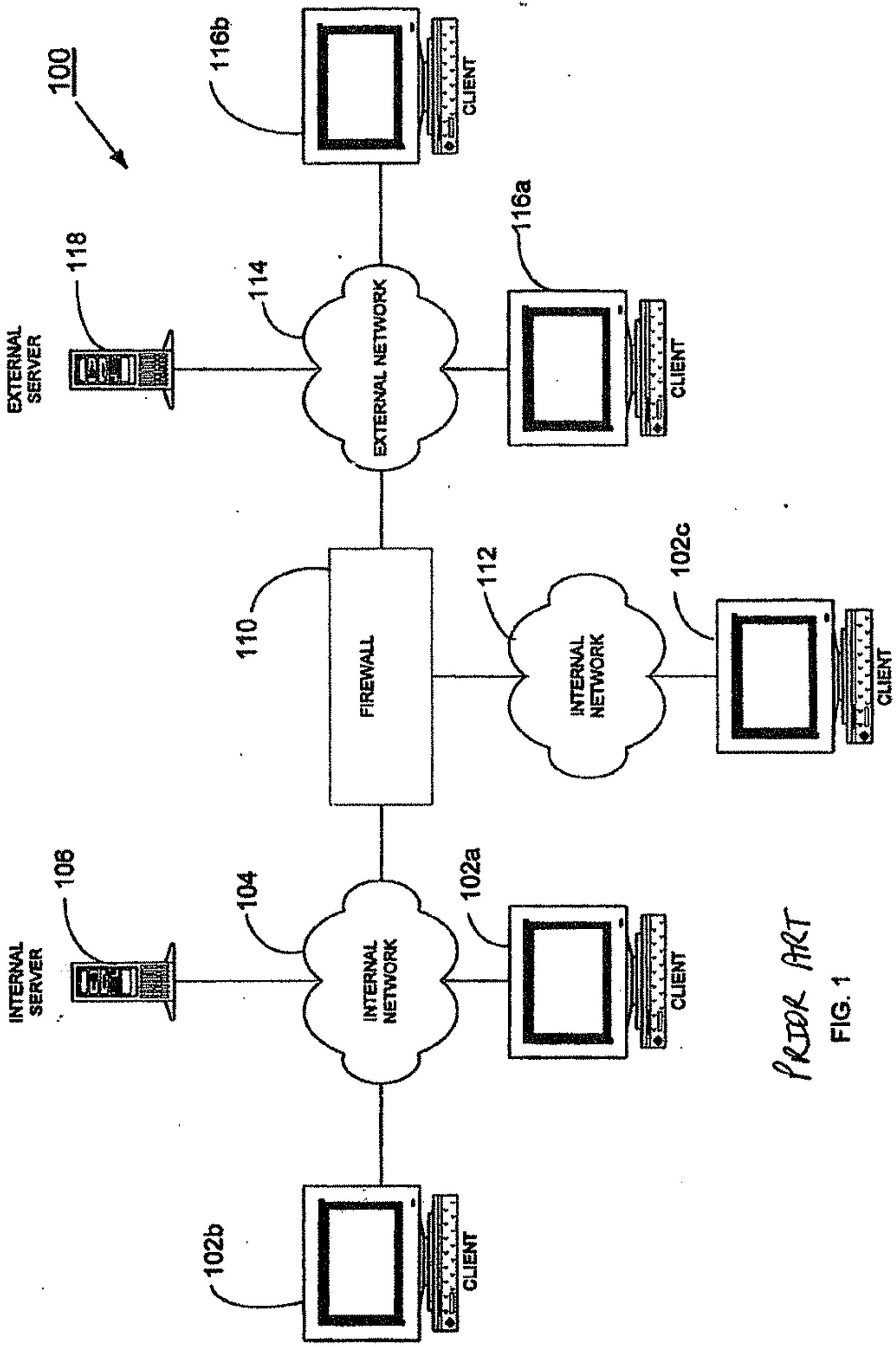
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

This responds to the Office Action mailed on September 10, 2003. Please amend the above-identified patent application as follows.

IN THE DRAWINGS

Corrected drawings are supplied herewith.

Enclosed is a copy of Figure 1 of the drawings showing the proposed amendment adding the label "PRIOR ART" to Figure 1 in red ink. Figures 4 and 5 illustrate aspects of the present invention; it would not be proper to add the "PRIOR ART" label to these figures.



PRIOR ART

FIG. 1

IN THE SPECIFICATION

The paragraph beginning at page 4, line 4 is amended as follows:

Today, a new protocol, lightweight directory access protocol (LDAP), is gaining wide acceptance in business. The LDAP standard defines an information model for a directory, a namespace for defining how directory information is referenced and organized, and a network protocol for accessing information in the directory. LDAP can also include an application programming interface (API). The LDAP protocol mandates how client and server computers can communicate with a LDAP directory. However, LDAP does not mandate how data should be stored. ~~LDAP directories are described in greater detail in "Introduction to Directories and the Lightweight Directory Access Protocol," available at http://www.stanford.edu/~hodges/talks/mactivity.ldap.97/index2.html.~~ More and more companies today use an LDAP directory server to store a database of employees. The LDAP directory generally can store an employee name, phone number, address and other information about the employee, and a password for modifying the employee's information.

A1

IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A system for authorizing client access to a network resource, comprising:

a server having at least one directory that can be accessed using a network protocol, said at least one directory being configured to store information concerning an entity's organization; and

a firewall that is configured to intercept network resource requests from a plurality of client users, said firewall being operative to authorize a network resource request based upon a comparison of the contents of at least part of one or more entries in said at least one directory to an authorization filter, wherein saia authorization filter is generated based on a directory schema that is predefined by said entity.

A2

2. (Original) The system of claim 1, wherein said at least one directory is a lightweight directory access protocol directory.

3. (Original) The system of claim 1, wherein said authorization filter is specified using a graphical user interface.

4. (Original) The system of claim 1, wherein said authorization filter implements a per-user authentication scheme.

5. (Original) The system of claim 1, wherein said authorization filter implements a per-service authentication scheme.

6. (Original) The system of claim 1, wherein said firewall and said directory communicate using secure socket layer communication.

7. (Original) The system of claim 1, wherein said firewall is configured to query multiple directories.

8. (Original) An authentication method at a firewall, comprising the steps of:

(a) receiving a network resource request from a client user;

(b) querying, using a network protocol, at least one directory that is configured to store information concerning an entity's organization, wherein said query is based upon an authorization filter that is generated based on a directory schema that is predefined by said entity;

(c) determining, based on the results of said query, whether the contents of at least part of one or more entries in said at least one directory satisfy said authorization filter; and

(d) permitting said network resource request through said firewall if said authorization filter is satisfied.

A2

9. (Original) The method of claim 8, wherein step (b) comprises the step of querying said at least one directory using a lightweight directory access protocol.

10. (Original) The method of claim 8, further comprising the step of specifying an authorization filter using a graphical user interface.

11. (Original) The method of claim 10, wherein said specifying step comprises the step of specifying an authorization filter that implements a per-user authentication scheme.

12. (Original) The method of claim 10, wherein said specifying step comprises the step of specifying an authorization filter that implements a per-service authentication scheme.

13. (Original) The method of claim 8, wherein step (b) comprises the step of querying said directory using secure socket layer communication.

14. (Original) The method of claim 8, wherein step (b) comprises the step of querying multiple directories.

15. (Original) The method of claim 8, wherein step (a) comprises the step of receiving a network resource request from a client user at an internal network.

16. (Original) The method of claim 8, wherein step (a) comprises the step of receiving a network resource request from a client user at an external network.

A2

17. (Original) A computer program product for enabling a processor in a computer system to implement an authentication process, said computer program product comprising:

a computer usable medium having computer readable program code embodied in said medium for causing a program to execute on the computer system, said computer readable program code comprising:

first computer readable program code for enabling the computer system to receive a network resource request from a client user;

second computer readable program code for enabling the computer system to query, using a network protocol, at least one directory that is configured to store information concerning an entity's organization, wherein said query is based upon an authorization filter that is generated based on a directory schema that is predefined by said entity;

third computer readable program code for enabling the computer system to determine, based on the results of said query, whether the contents of at least part of one or more entries in said at least one directory satisfy said authorization filter; and

fourth computer readable program code for enabling the computer system to permit said network resource request through said firewall if said authorization filter is satisfied.

REMARKS

Applicant has carefully reviewed and considered the Office Action mailed on September 10, 2003, and the references cited therewith.

Claim 1 was amended. No new claims were added. Claims 1-17 remain pending in this application.

§101 Rejection of the Claims

Claims 1-7 were rejected under 35 USC § 101 as being directed to non-statutory matter. Claim 1 has been amended to include a reference to a server where the directory is stored.

§103 Rejection of the Claims

Claims 1-17 were rejected under 35 USC § 103(a) as being unpatentable over Reid (U.S. Patent No. 6,131,120) in view of "The Microsoft Computer Dictionary, 1997", further in view of "Check Point Account Management Client, Version 1.0, 1998".

Reid describes a network security protocol in which router/gateways are used to control network traffic passing through each router/gateway.

> An enterprise directory residing on a directory server stores the names, workstations, router/gateways, servers, IP addresses locations, passwords, and encryption keys for individuals. Periodically, the directory server downloads to each router/gateway across the WAN router/gateway access lists (RALs), thereby controlling all network access across the WAN. Also periodically, the directory server downloads user control files (UCFs) to servers in the network, thereby controlling all server access across the WAN. This directory-based invention thus provides enhanced network control, and enhanced network security.

Reid, col. 6, lines 2-12. Reid, therefore, controls access to data within a network by limiting traffic through router/gateways (using a RAL at each router/gateway) and by limiting access to files within servers (using a UCF at each server). Both the RAL and the UCF are generated by a directory server and distributed periodically to their respective router/gateways and servers.

In addition, one or more of the servers can provide user authentication. Reid states that authentication at the server level is superior to that at the firewall since "distributed authentication provides greatly enhanced security over a firewall-protected network." Reid, col. 6, lines 63-65.

Applicant teaches that it can be difficult to maintain a directory for computer security at the same time that one is maintaining a directory for other purposes. As noted at p. 4, lines 16-23, maintenance of a firewall authentication database is especially burdensome in companies with a large amount of employee turnover or in companies with a large number of firewalls. Applicant teaches that it can be advantageous to configure the firewall to leverage existing databases, such as an LDAP server storing employee information such as is shown in Fig. 4.

As is described on p. 9, line 1 through p. 10, line 2 and as is shown in Fig. 3, an authorization module 206 within firewall 110 receives a request from a user to access an application or resource on the other side of firewall 110. Authorization module 206 authenticates the user, and then determines whether that user is authorized to have his access request fulfilled by querying a server 106 having an LDAP directory. The entry read from the LDAP directory that is associated with the user is compared to an authorization filter. If one or more attributes of the entry does not satisfy the filter, the user is not authorized to access the requested application or resource and the request fails. If, however, all the attributes of the entry satisfy the filter, the user is authorized to access the requested application or resource and the request is allowed through firewall 110.

Claim 1, as amended, includes a server having at least one directory and a firewall configured to intercept network resource requests from a plurality of client users. The firewall is "operative to authorize a network resource request based upon a comparison of the contents of at least part of one or more entries in said at least one directory to an authorization filter, wherein said authorization filter is generated based on a directory schema that is predefined by said entity."

The Examiner stated that since the router/gateway access list is sent to each router/gateway and controls who may have access through router/gateway,

> The examiner asserts that in order for the directory to generate the appropriate access list, each router/gateway must have transmitted its access criteria to the directory. The examiner further asserts that this criteria is an authorization filter and that in order for the directory to send back a correct access list, some comparison must have been made with directory entries and the router/gateway criteria (authorization filter).

Office Action, p. 5, third full paragraph. Applicant disagrees.

As noted above, Reid makes it clear (col. 6, lines 2-12) that all security information is stored within an enterprise directory and that periodically that directory downloads RALs and UCFs to router/gateways and servers, respectively. "Because the directory knows the location and IP address of each user, and the location and IP address of each router/gateway, a directory application can periodically populate the RAL in each router/gateway on the network using LDAP. Entries in the directory thereby control the entire network and the network router/gateway configuration management is automated." Col. 6, lines 19-25. Therefore, despite the Examiner's statement that "in order for the directory to send back a correct access list, some comparison must have been made with directory entries and the router/gateway criteria (authorization filter)," no such authorization filter is provided in Reid. At least one limitation of claims 1-7 is, therefore, not present in any of the references cited by the Examiner. Reconsideration of claims 1-7 is respectfully requested.

Claim 8 is to an authentication method which determines if a user is authorized to access an application or a resource by applying an authorization filter to an entry associated with the user stored in a directory. Once again, Reid does not describe the application of an authorization filter to an entry read from a directory. At least one limitation of claims 8-16 is, therefore, not present in any of the references cited by the Examiner. Reconsideration of claims 8-16 is respectfully requested.

Claim 17 is a computer program product which includes program code for applying an authorization filter to an entry associated with the user stored in a directory. Once again, Reid does not describe the application of an authorization filter to an entry read from a directory. At least one limitation of claim 17 is, therefore, not present in any of the references cited by the Examiner. Reconsideration of claim 17 is respectfully requested.

Claims 6 and 13 were rejected under 35 USC § 103(a) as being unpatentable over Reid as applied to 1 and 8 above, and further in view of Elgamal (U.S. Patent No. 5,657,390).

Neither Reid nor Elgamel teach the use of an authorization filter as described by Applicant. In addition, Elgamal does not describe the use of a secure socket layer communication to distribute an entry in an LDAP database as taught by Applicant and claimed in claims 6 and 13. Reconsideration of claims 6 and 13 is respectfully requested.

Claims 3 and 10 were rejected under 35 USC § 103(a) as being unpatentable over Reid as applied to claims 1 and 8 above, and further in view of Wesinger (U.S. Patent No. 5,898,830).

Neither Reid nor Wesinger teach the use of an authorization filter as described by Applicant. In addition, Wesinger does not describe the use of GUI to specify how to implement the authorization filer as taught by Applicant and claimed in claims 3 and 10. Reconsideration of claims 3 and 10 is respectfully requested.

Claims 11 and 12 were rejected under 35 USC § 103(a) as being unpatentable over Reid in view of Wesinger as applied to claims 3 and 10 above, and further in view of Reid as applied to claims 4 and 5 above.

Neither Reid nor Wesinger teach the use of an authorization filter as described by Applicant. In addition, Wesinger does not describe the methods used to implement the authorization filer as taught by Applicant and claimed in claims 11 and 12. Reconsideration of claims 11 and 12 is respectfully requested.

Conclusion

Applicant respectfully submits that the claims are in condition for allowance and notification to that effect is earnestly requested. The Examiner is invited to telephone Applicant's attorney at (612) 373-6909 to facilitate prosecution of this application.

If necessary, please charge any additional fees or credit overpayment to Deposit Account No. 19-0743

Respectfully submitted,

Thomas D. Ashoff, et al.

By their Representatives,

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. Box 2938
Minneapolis, MN 55402
(612) 373-6909

Date January 12, 2004 By Thomas F. Brennan
Thomas F. Brennan
Reg. No. 35,075

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: Commissioner of Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on this 12 day of January, 2004.

GINA OPHUS

Name

Gina Ophus

Signature

JC542 U.S. PTO
09/024576
02/17/98

ISSUE CLASSIFICATION	
Class	Subclass

[redacted]

PATENT NUMBER
6357010
6357010

U.S. UTILITY PATENT APPLICATION

O.I.P.E. SCANNED CG Q.A.	PATENT DATE MAR 1 2 2002

SECTOR	CLASS	SUBCLASS	ART UNIT	EXAMINER
	39 713	201	21	

FILED WITH: □ DISK (CRF) □ FICHE
(Attached in pocket on right inside flap)

CERTIFICATE
MAR 2 5 2003
OF CORRECTION

PREPARED AND APPROVED FOR ISSUE

ISSUING CLASSIFICATION

ORIGINAL		CROSS REFERENCE(S)						
CLASS	SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)					
713	201	709	225					
INTERNATIONAL CLASSIFICATION		713	200					
G06F	12/14							
G06F	15/173							
H04L	12/66							
H04L	9/00							
	/			□ Continued on Issue Slip Inside File Jacket				

□ TERMINAL DISCLAIMER	DRAWINGS			CLAIMS ALLOWED	
	Sheets Drwg.	Figs. Drwg.	Print Fig.	Total Claims	Print Claim for O.G.
	7	7	2	37	1

□ a) The term of this patent subsequent to ____________ (date) has been disclaimed.

(Assistant Examiner) 8/22/01 (Date)

NOTICE OF ALLOWANCE MAILED
8-24-01

□ b) The term of this patent shall not extend beyond the expiration date of U.S Patent. No. ____________

THOMAS LEE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

(Primary Examiner) 8/20/01 (Date)

ISSUE FEE J.L.G.

Amount Due	Date Paid
620.00	11-26-01

□ c) The terminal ____ months of this patent have been disclaimed.

(Legal Instruments Examiner) 8/01 (Date)

ISSUE BATCH NUMBER
K97

WARNING:
The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368. Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.

Form PTO-436A
(Rev. 10/97)

ISSUE FEE IN FILE

Formal Drawings (___ shts) set

(LABEL AREA)

(FACE)

✓ $2787
#17

IN THE UNITET STATES PATENT AND TRADEMARK OFFICE

Applicant:	Richard R. Viets et al.
Title:	SYSTEM AND METHOD FOR CONTROLLING ACCESS TO DOCUMENTS STORED ON AN INTERNAL NETWORK

Docket No.:	105.140US1	Serial No.: 09/024,576
Filed:	February 17, 1998	Due Date: July 17, 2001
Examiner:	Nguyen Xuan Nguyen	Group Art Unit: 2787

Commissioner for Patents
Washington, D.C. 20231

We are transmitting herewith the following attached items (as indicated with an "X"):

X A return postcard.
X Amendment and Response Under 37 CFR 1.111 (6 Pages).
X Petition for Extension of Time (1 pg.)
X A check in the amount of $445.00 to cover the Extension of Time Fee.

Please consider this a PETITION FOR EXTENSION OF TIME for sufficient number of months to enter these papers and please charge any additional required fees or credit overpayment to Deposit Account No. 19-0743.

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. Box 2938, Minneapolis, MN 55402 (612-373-6900)

By: Michael L. Schwegman
Atty: Micheal L. Schwegman
Reg. No. 25,816

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: Commissioner for Patents, Washington, D.C. 20231, on this 17th day of July, 2001.

PATRICIA A. HULTMAN
Name

Patricia A. Hultman
Signature

Customer Number 21186
SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A. P.O. Box 2938, Minneapolis, MN 55402 (612-373-6900)
(GENERAL)

#17
paper
8-2-01

S/N 09/024,576 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Richard R. Viets et al. Examiner: Nguyen Xuan Nguyen
Serial No.: 09/024,576 Group Art Unit: 2787
Filed: February 17, 1998 Docket: 105.140US1
Title: SYSTEM AND METHOD FOR CONTROLLING ACCESS TO DOCUMENTS STORED ON AN INTERNAL NETWORK

AMENDMENT AND RESPONSE UNDER 37 CFR § 1.111

Commissioner for Patents
Washington, D.C. 20231

Applicant has reviewed the Office Action mailed on January 17, 2001.

This response is accompanied by a Petition, as well as the appropriate fee, to obtain a three-month extension of time for the period for responding to the Office action, thereby moving the deadline for response from April 17, 2001 to July 17, 2001.

REMARKS

Applicant has carefully reviewed and considered the Office Action mailed on January 17, 2001, and the references cited therewith.

§103 Rejection of the Claims

Claims 1, 2, 4, 5 and 35 were rejected under 35 USC § 103(a) as being unpatentable over Hudson (U.S. Patent No. 6,055,637) in view of Kitain (U.S. Patent No. 5,864,871).

As the Examiner suggests, Hudson describes a role-based resource access control system which relies on temporary user credential tokens to control access to resources. Each token preferably contains verification information used to verify the identity of the user and authorization information used to determine if the user is authorized to access the selected object. Authorization information may define organizational groups, functional groups, or roles.

As the Examiner notes, each token may also include security rules defining the access permissions associated with particular resources. The access permissions are generated "on the fly".

> Constructed in this manner, there is no need to write and store security controls for each individual user that would be stored and remain in the language of the security package in its database. Instead, resource access control system and method of the present invention communicate with the security package of a resource only when the user desires access thereto and dynamically generate the

> security rule, or temporary access permission for the user's access for each session. The security permission is removed at the end of the user's session, so that there is no update required when that user no longer needs access to that particular resource or change responsibility, function, or status with the company. The security permission is also removed when the session did not end in a normal manner to prevent unauthorized access.

Hudson, col. 4, lines 7-21.

The Examiner stated that Hudson teaches each of the limitations of claims 1 and 35 except for "fetching the requested document as a proxy and sending the requested document to the client." Applicant respectfully disagrees. Hudson does not teach "building a document list naming documents available to clients assigned to the client's role" as taught by Applicant and claimed in claims 1-37. Instead, Hudson describes a system which

> communicate[s] with the security package of a resource only when the user desires access thereto and dynamically generate[s] the security rule, or temporary access permission for the user's access for each session. The security permission is removed at the end of the user's session, so that there is no update required when that user no longer needs access to that particular resource or change responsibility, function, or status with the company.

Hudson, col. 4, lines 7-21. There is no "building a document list" as defined and claimed by Applicant. Instead, the list of security rules is more of an audit trail of the resources to which the user was granted access during the session than a list of permitted documents. That is, Hudson's access permissions are determined on "on the fly" and erased when the session ends.

In addition, Hudson does not teach "determining if the requested document is on the list of documents" as taught by Applicant and claimed in claims 1-22 and 35. The Examiner implied that Hudson's "security rules" are equivalent to Applicant's "document list" and that the process of communicating with the security package of a resource is the same as consulting a document list. Applicant respectfully disagrees. Even after a security rule has been calculated for access to a resource, subsequent accesses to that resource during that particular session resembles queries to an access control list rather than access to a list of permitted documents.

In addition, Hudson teaches away from the present invention. Hudson's preferred system presents a list of resources to which the user is allowed access as part of the authentication process. *Id.*, col. 5, lines 31-35 The user is, therefore, prevented from accessing resources to which he or she has no access rights.

Kitain describes a web server coupled to a central repository server. The central repository server is, in turn, coupled to one or more databases of information. Access by a user to the central repository server is controlled through the web server, which authenticates the user. Once authenticated, the central repository server determines the information the user is authorized to receive. Different users may be permitted to access different subsets of the documents stored at the central repository. Users can institute searches of the central repository server; the documents found by the search are screened and only documents the user has permission to access are displayed in the resulting document list. *Kitain*, col. 5, line 7 to col. 6, line 38.

The Examiner stated that Kitain performs the proxy function which is a limitation of claims 1-6 and 35. There is, however, no discussion in Kitain of the use of a proxy function to hide the address of the document sought.

For the above reasons, Applicant respectfully submits that the Examiner has failed to establish a *prima facie* case of obviousness in the rejection of claims 1, 2, 4, 5, and 35 and requests reconsideration and allowance of all claims.

In the rejection of claims 2 and 5, the Examiner stated that although Hudson does not teach the use of a URL for each document page, it would be obvious to employ Kitain's teaching of the use of URLs as addresses for documents to build Hudson's security rules. Applicant respectfully submits that there is no teaching in either Hudson or Kitain for developing a security rule based on the URL of a desired resource. The Examiner is, therefore, taking official notice of such a teaching. Applicant respectfully requests that the Examiner produce a document in support of such a teaching. In the absence of such a document and for the reasons stated for claims 1-37 above, Applicant respectfully submits that the Examiner has failed to establish a *prima facie* case of obviousness in the rejection of claims 2 and 5 and requests reconsideration and allowance of all claims.

Claims 3 and 6 were rejected under 35 USC § 103(a) as being unpatentable over Hudson (U.S. Patent No. 6,055,637) in view of Kitain (U.S.Patent No. 5,864,871) and in further view of Kiernan et al. (U.S. Patent No. 5,701,137).

Hudson and Kitain were discussed above.

Kiernan describes a system for displaying a graphical tree structure in a windowing environment. Since claims 3 and 6 include the limitations of claims 1 described above and since claim 1 distinguishes over the combination of Hudson and Kitain, claims 3 and 6 distinguish over the references cited by the Examiner for the reasons discussed above. Applicant reserves the right to further distinguish over Kiernan as necessary.

Claims 7, 9-11, 13, 14, 17, 21-23, 25-30, 32-34, 36 and 37 were rejected under 35 USC § 103(a) as being unpatentable over Hudson (U.S. Patent No. 6,055,637) in view of Logan (U.S. Patent No. 5,802,299).

Hudson is discussed above.

Logan describes a hypertext display system which includes "access control programs for analyzing and rewriting the text found in accessed HTML pages before those pages are displayed and perform predetermined functions defined by stored access control information when the user activates selected links." *Logan*, col. 5, lines 15-19. The access control information may be created using, for instance, a remote authoring computer before being downloaded to a kiosk.

If data is accessed across the Internet,

> When the returned displayable data is an HTML document, the text of that document is processed by the access control mechanism 110 which includes a mechanism 130 for rewriting the HTML page in accordance with information in a string list data structure 133. The string list 133 typically contains a collection of text replacement request commands each including of a designated target string and a designated replacement string. Whenever one of the target strings in the structure 133 is found within the text of an incoming HTML document, that target string is replaced by the associated replacement string before the incoming HTML document is displayed by the web browser program 107.
>
> The HTML text replacement function performed at 130 in the access control mechanism 110 may be used to provide a number or useful functions. In addition to rewriting displayable text, the rewriting mechanism 130 may add new links to additional information which the kiosk owner may wish to communicate to the kiosk user, may delete links to information which should be hidden to the user, or may substitute replacement links. Unlike the URL transition display generating mechanism 113, which is capable of inserting one or more display pages before a page designated by the URL request 109, the mechanism 130 may be used to substitute a different target page for the page specified by a link imbedded in an incoming HTML document, and may also be used to eliminate the highlighting of, or rewrite, the displayed anchor text which is associated with the linked URL in the HTML page. The string list 133 includes a collection of target+replacement string pairs. The mechanism 130 searches the HTML page fetched by the access mechanism 120, searching for a match to each of the target strings, and when found substitutes the replacement string for the target string.

Logan, col. 6, line 59 - col. 7, line 25.

Logan's approach requires detailed lists of target and replacement strings. It is not based on whether the user has permission to access the linked document. In addition, Logan does not teach a document processor as part of the document control server. Instead, the link rewriting is performed at the destination. For these reasons; claims 7, 9-11, 13, 14, 17, 21-23, 25-30, 32-34, 36 and 37 distinguish over the combination of Hudson and Logan.

Claims 12 and 15 were rejected under 35 U.S.C. 103(a) as being unpatentable over Hudson (U.S. Patent No. 6,055,637) in view of Logan (U.S. Patent No. 5,802,299) and in further view of Kiernan et al. (U.S. Patent No.5,701,137).

Claims 12 and 15 are dependent on and include the limitations of claim 11. Claims 12 and 15, therefore, distinguish over the combination of Hudson and Logan for the reasons discussed above. Applicant reserves the right to further distinguish over Kiernan as necessary.

Claim 19 was rejected under 35 USC § 103(a) as being unpatentable over Hudson (U.S. Patent No. 6,055,637) in view of Logan (U.S. Patent No. 5,802,299) and in further view of Dustan et al. (U.S. Patent No.5,884,312).

Claim 19 is dependent on and includes the limitations of claim 7. Claim 19, therefore, distinguishes over the combination of Hudson and Logan for the reasons discussed above. Applicant reserves the right to further distinguish over Dustan as necessary.

Claims 8, 20, 18, and 16 were rejected under 35 USC § 103(a) as being unpatentable over Hudson (U.S..Patent No. 6,055,637) in view of Logan (U.S. Patent No. 5,802,299) and in further view of Gore Jr. et al. (U.S. Patent No. 5,826,029).

Claims 8, 18 and 20 are dependent on and include the limitations of claim 7. Claims 8, 18 and 20, therefore, distinguish ovei the combination of Hudson and Logan for the reasons discussed above. Applicant reserves the right to further distinguish over Gore as necessary.

Claim 16 is dependent on and includes the limitations of claim 7. Claim 16, therefore, distinguishes over the combination of Hudson and Logan for the reasons discussed above. Applicant reserves the right to further distinguish over Gore as necessary.

Claims 24 and 31 were rejected under 35 U.S.C. 103(a) as being unpatentable over Hudson (U.S. Patent No. 6,055,637) in view of Logan (U.S. PATENT No. 5,802,299) and in further view of Kitain et al. (U.S. Patent No.5,864,871).

Claim 24 is dependent on and includes the limitations of claim 23. Claim 24, therefore, distinguishes over the combination of Hudson and Logan for the reasons discussed above. Applicant reserves the right to further distinguish over Kitain as necessary.

Claim 31 is dependent on and includes the limitations of claim 30. Claim 31, therefore, distinguishes over the combination of Hudson and Logan for the reasons discussed above. Applicant reserves the right to further distinguish over Kitain as necessary.

Conclusion

Applicant respectfully submits that the claims are in condition for allowance and notification to that effect is earnestly requested. The Examiner is invited to telephone Applicant's attorney (612-373-6909) to facilitate prosecution of this application.

If necessary, please charge any additional fees or credit overpayment to Deposit Account No. 19-0743.

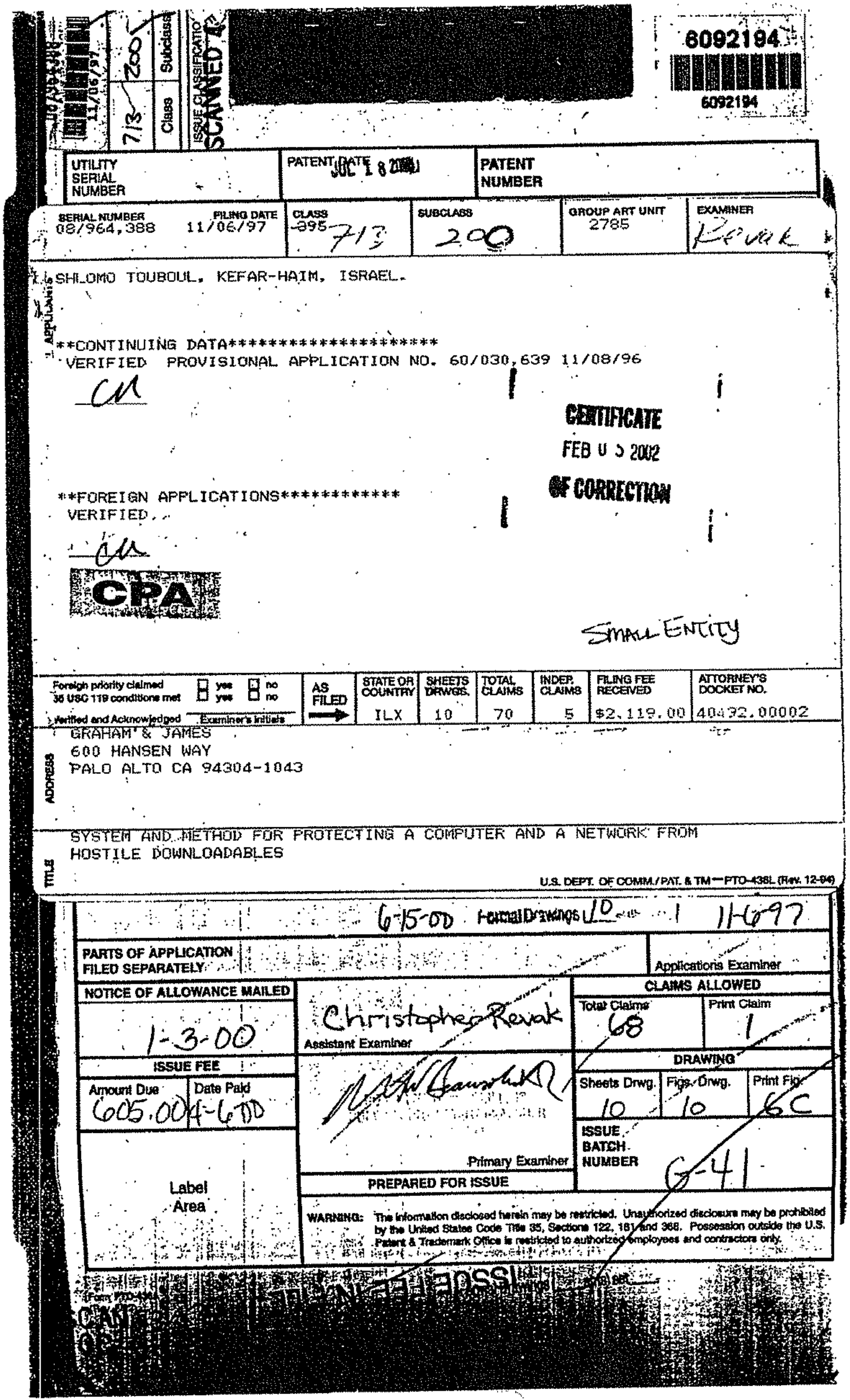
Respectfully submitted,
RICHARD R. VIETS ET AL.
By their Representatives,
SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. Box 2938
Minneapolis, MN 55402
(612) 373-6909

Dte. Name 7-17-01 By M. 22159
Micheal L. Schwegman
Reg. No. 25,816

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: Commissioner of Patents, Washington, D.C. 20231, on this 17th day of July, 2001.

PATRICIA A. HULTMAN
Name

Signature



11/06/97

Class	Subclass
713	200

ISSUE CLASSIFICATION

SCANNED

[redacted]

6092194

6092194

UTILITY SERIAL NUMBER	PATENT DATE JUL 18 2000	PATENT NUMBER

SERIAL NUMBER	FILING DATE	CLASS	SUBCLASS	GROUP ART UNIT	EXAMINER
08/964,388	11/06/97	~~395~~ 713	200	2785	Revak

APPLICANTS

SHLOMO TOUBOUL, KEFAR-HAIM, ISRAEL.

CONTINUING DATA*******************

VERIFIED PROVISIONAL APPLICATION NO. 60/030,639 11/08/96

CM

CERTIFICATE

FEB 05 2002

OF CORRECTION

FOREIGN APPLICATIONS**********

VERIFIED.

CM

CPA

Small Entity

Foreign priority claimed □ yes □ no 35 USC 119 conditions met □ yes □ no Verified and Acknowledged ____ Examiner's initials	AS FILED →	STATE OR COUNTRY	SHEETS DRWGS.	TOTAL CLAIMS	INDEP. CLAIMS	FILING FEE RECEIVED	ATTORNEY'S DOCKET NO.
		ILX	10	70	5	$2,119.00	40432.00002

ADDRESS

GRAHAM & JAMES
600 HANSEN WAY
PALO ALTO CA 94304-1043

TITLE

SYSTEM AND METHOD FOR PROTECTING A COMPUTER AND A NETWORK FROM HOSTILE DOWNLOADABLES

U.S. DEPT. OF COMM./PAT. & TM—PTO-436L (Rev. 12-94)

6-15-00 Formal Drawings 10 1 11-6-97

PARTS OF APPLICATION FILED SEPARATELY

Applications Examiner

NOTICE OF ALLOWANCE MAILED

1-3-00

Christopher Revak
Assistant Examiner

Primary Examiner

PREPARED FOR ISSUE

ISSUE FEE

Amount Due	Date Paid
605.00	4-6-00

CLAIMS ALLOWED

Total Claims	Print Claim
68	1

DRAWING

Sheets Drwg.	Figs./Drwg.	Print Fig.
10	10	6C

ISSUE BATCH NUMBER G-41

Label Area

WARNING: The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368. Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.

ISSUE FEE

(Form PTO-436A)

#15

EXPRESS MAIL LABEL NO: ELx.36875714US

Our Docket No. 40492.00002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

#14/B

In Re Application of:
Shlomo Touboul

Serial No.: 08/964,388

Filed: November 6, 1997

For: System and Method for Protecting a Computer and a Network from Hostile Downloadables

Examiner: B. Shaw

Art Unit: 2785

11/2/99

BOX - CPA
Assistant Commissioner for Patents
Washington, D.C. 20231

PRELIMINARY AMENDMENT

Sir:

This preliminary amendment addresses the rejections of the final office action mailed on June 17, 1999, the statutory period for response in the parent application being extended until November 17, 1999 by the enclosed 2-month petition for extension of time. Before considering the subject application, please amend it as follows:

IN THE SPECIFICATION:

On page 11 line 14, after "Downloadable," delete "the".

IN THE ABSTRACT:

Please remove the paragraph break.

EXPRESS MAIL LABEL NO: EL038875714US

Our Docket No. 40492.00002

IN THE CLAIMS:

1. (Twice amended) A computer-based method, comprising the steps of:

receiving an incoming Downloadable addressed to a client, by a server that serves as a gateway to the client;

B1 Ins c1 comparing, by the server, [at least a portion of] Downloadable security profile data pertaining to the Downloadable c1 against a security policy to determine if the security policy has been violated; and

preventing execution of the Downloadable by the client if the security policy has been violated.

B2 2. (Once amended) The method of claim 1, further comprising the step[s] of decomposing the Downloadable into the Downloadable security profile data[, and comparing the Downloadable security profile data against the security policy].

~~31~~ 32. (Twice amended) A system for execution by a server that serves as a gateway to a client, the system comprising:

a security policy;

an interface for receiving an incoming Downloadable addressed to a client;

B3 Ins c2 a comparator, coupled to the interface, for [applying] comparing Downloadable security profile data pertaining to the Downloadable c2 against the security policy [to the Downloadable] to determine if the security policy has been violated; and

a logical engine for preventing execution of the Downloadable by the client if the security policy has been violated.

B4 ~~53~~ 54. (Once amended) The system of claim ~~31~~ 32, further comprising a code scanner coupled to the comparator for decomposing the Downloadable into the Downloadable security profile data.

EXPRESS MAIL LABEL NO: EL058875714US

Our Docket No. 40492.00002

64 ~~63~~. (Twice amended) A system for execution on a server that serves as a gateway to a client, comprising:

means for receiving an incoming Downloadable addressed to a client;

IS e3 means for comparing Downloadable security profile data pertaining to the Downloadable, e3 against a security policy to determine if the security policy has been violated; and

means for preventing execution of the Downloadable by the client if the security policy has been violated.

65 ~~64~~. (Twice amended) A computer-readable storage medium storing program code for causing a server that serves as a gateway to a client to perform the steps of:

β5 receiving an incoming Downloadable addressed to a client;

comparing Downloadable security profile data pertaining to the Downloadable against a security policy to determine if the security policy has been violated; and ✓

1 preventing execution of the Downloadable by the client if the security policy has been violated.

65. (Twice amended) A computer-based method for generating a Downloadable ID to identify a Downloadable, comprising the steps of:

obtaining a Downloadable;

fetching, if the Downloadable includes one or more references to a component, at least one component identified by the one or more references; and

performing a function on the Downloadable and all components fetched to generate a Downloadable ID.

β6 70. (Twice amended) The method of claim [67] 65; wherein the step of fetching includes fetching all components referenced by the Downloadable.

Please add the following claims:

31~~77~~. The method of claim 1, further comprising the steps of recognizing the incoming Downloadable, and obtaining the Downloadable security profile data for the incoming Downloadable from memory.

β?

63 ~~78~~. The system of claim 32 ~~31~~, further comprising memory storing the Downloadable security profile data for the incoming Downloadable.--

REMARKS

Claims 1-22, 26-59, 61-66 and 68-76 were examined and rejected in this case. Claims 1, 2, 31, 53, 63-65 and 70 are being amended. Claims 77 and 78 are being added. Claims 1-22, 26-59, 61-66 and 68-78 are currently pending. Reconsideration of the application as amended is respectfully requested.

Applicant requests the Examiner to enter the above amendment to the Specification. No new matter is being added.

In paragraph 1, the Examiner objected to the Abstract because it was written as two paragraphs. Accordingly, Applicant is amending the Abstract to remove the paragraph break.

In paragraph 2, the Examiner rejected claim 70 for being dependent on a canceled claim. Accordingly, Applicant is amending claim 70 to depend from pending claim 65.

In paragraph 4, the Examiner rejected claims 1, 3, 7-10, 31-35, 55, 56, 63, 64 and 74 under 35 USC § 103(a) as being unpatentable McManis in view of Boebert. McManis teaches examining digital signatures by the originating party, by the compiling party, and by the executing party. Boebert teaches secure transfer of data through a secure computer to clients. However, claims 1, 31, 63, 64 and 74, as amended, similarly recite "comparing, by the server, Downloadable security profile data pertaining to the Downloadable against a security policy to determine if the security policy has been violated." On page 12 lines 17-20, the Specification describes an embodiment wherein the particular Downloadable security profile data 310 includes "the list of all potentially hostile or suspicious computer operations that may be attempted by a specific Downloadable 307, and may also include the respective arguments of these operations." Applicant respectfully submits that neither McManis nor Boebert teaches this step of comparing potentially hostile operations in a Downloadable against a security policy as recited in independent claims 1, 31, 63, 64 and 74. For at least the same reasons, Applicant respectfully submits that claims 3, 7-10, 32-35, 55, 56, dependent therefrom, are also patentable.

In paragraph 5, the Examiner rejected claims 2, 4-6, 11-14, 16-22, 29, 30, 36-46, 52-54, 57-61, 65, 66, 68, 69, 75 and 76 under 35 USC § 103(a) as unpatentable over McManis in view of Boebert and further in view of Deo. In subparagraph 5 of paragraph 5, the Examiner cited that any "user" identification would be obvious over McManis in view of Chang. Since the overall rejection is based on McManis, Boebert and Deo, Applicant is unsure whether the Examiner intended to cite Boebert instead of Chang. For the purposes of this response, Applicant will assume that the Examiner intended to cite Boebert. Applicant requests clarification.

Deo teaches a security platform using object-oriented rules for UNIX operating systems. As stated above, independent claims 1, 31, 63, 64 and 74, as amended, similarly recite "comparing, by the server, Downloadable security profile data pertaining to the Downloadable against a security policy to determine if the security policy has been violated." Independent claim 76 also includes this limitation. Like McManis and Boebert, Applicant respectfully submits that Deo does not teach this step. Accordingly, for at least these reasons, Applicant respectfully submits that claims 2, 4-6, 11-14, 16-22, 29, 30, 36-46, 52-54, 57-61 and 75, dependent therefrom, and independent claim 76 are patentable over McManis in view of Boebert and further in view of Deo.

Independent claim 65 recites a method for generating a Downloadable ID to identify the incoming Downloadable. Neither McManis, Boebert nor Deo teach this method. Accordingly, Applicant respectfully submits that claim 65, and for at least this reason claims 66, 68, 69, dependent therefrom, are patentable over McManis in view of Boebert and further in view of Deo.

In paragraph 6, the Examiner rejected claims 15 and 62 under 35 USC § 103(a) as unpatentable over McManis in view of Boebert and further in view of Official Notice. The Examiner asserted that event logs are known in the art. However, since claims 15 and 62 depend from claims 1 and 31, respectively, Applicant respectfully submits that claims 15 and 62 are patentable for at least the same reasons.

In paragraph 7, the Examiner rejected claims 26-28, 50, 51 and 71-73 under 35 § 103(a) as unpatentable over McManis in view of Boebert and further in view of Ji. The Examiner admitted that McManis and Boebert do not teach hostile Downloadable detection, and asserted that Ji teaches it. Applicant respectfully traverses. Ji teaches gateway detection of viruses attached to executable files, and does not teach hostile Downloadable detection. As is well known in the art, a Downloadable is mobile code that is requested by an ongoing process, downloaded from a source computer to a destination computer for automatic execution. The programs or documents of Ji are not Downloadables. Further, Ji does not teach a "comparing, by the server, Downloadable security profile data pertaining to the Downloadable against a security policy to determine if the security policy has been violated," as recited in independent claims 1 and 31. Since claims 26-28 and 71-73 depend from claim 1 and claims 50 and 51 depend from claim 31, Applicant respectfully submits that claims 26-28, 50, 51 and 71-73 are patentable for at least the same reasons.

In paragraph 8, the Examiner rejected claims 47-49 under 35 USC § 103(a) as unpatentable over McManis, in view of Boebert, further in view of Deo and still further in view of Ji. For at least the reasons discussed above, Applicant respectfully submits that claims 47-49 are patentable.

Applicant respectfully requests that the Examiner withdraw the rejection of claims 1-22, 26-59, 61-66 and 68-76.

EXPRESS MAIL LABEL NO: EL058875714US

Our Docket No. 40492.00002

If the Examiner has any questions or needs any additional information, the Examiner is invited to telephone the undersigned attorney at (650) 843-3392.

If for any reason an insufficient fee has been paid, the Assistant Commissioner is hereby authorized to charge the insufficiency to Deposit Account No. 05-0150.

Respectfully Submitted,
Shlomo Touboul

Dated: 10-27-99

Graham & James LLP
600 Hansen Way
Palo Alto, CA 94304-1043
650-856-6500

Marc A. Sockol
Attorney for Applicants
Reg. No. 40,823

ISSUE CLASSIFICATION	
Class	Subclass
713	181

PATENT NUMBER
6804780

U.S. UTILITY Patent Application

SV

O.I.P.E.	PATENT DATE
SCANNED Q.A. MC	OCT 1 2 2004

APPLICATION NO.	CONT/PRIOR	CLASS	SUBCLASS	ART UNIT	EXAMINER
09/539667	D	713	181	2785 2131	Revak

APPLICANTS [illegible]

2184

TITLE [illegible]

PTO-2040
12/99

ISSUING CLASSIFICATION

ORIGINAL		CROSS REFERENCE(S)						
CLASS	SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)					
713	181	713	201	176				
INTERNATIONAL CLASSIFICATION		717	178					
H04L	9/00							
G06F	11/30							
			□ Continued on Issue Slip Inside File Jacket					

9/10/04 [illegible] 10 [illegible] 3/30/00

☒ TERMINAL DISCLAIMER	DRAWINGS			CLAIMS ALLOWED	
	Sheets Drwg.	Figs. Drwg.	Print Fig.	Total Claims	Print Claim for O.G.
	10	10	8	18	1
□ The term of this patent subsequent to ____ (date) has been disclaimed.	Christopher Revak 5/13/04 (Assistant Examiner) (Date)			NOTICE OF ALLOWANCE MAILED 06-4-04	
☒ The term of this patent shall not extend beyond the expiration date of U.S Patent. No. 6,092,194	AYAZ SHEIKH SUPERVISORY PATENT EXAMINER TECHNOLOGY CENTER 2100 5/14/04 (Primary Examiner) (Date)			ISSUE FEE	
				Amount Due $ 665.00	Date Paid 9/3/04
□ The terminal ___ months of this patent have been disclaimed.	Brenda Harris 7/04 (Legal Instruments Examiner) (Date)			ISSUE BATCH NUMBER	

WARNING:
The information disclosed herein may be restricted Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368. Possession outside the U.S Patent & Trademark Office is restricted to authorized employees and contractors only.

Form PTO-436A
(Rev. 6/99)

FILED WITH: □ DISK (CRF) □ FICHE □ CD-ROM
(Attached in pocket on right inside flap)

(FACE)

#6A/8-9-03 v Jones

Attorney's Docket No.: 43426.00011 *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF MAILING

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on

Date: July 31, 2003

By: EM Janikowski
Eileen M. Janikowski

In Re Patent Application of:)
) Examiner: Christopher A. Revak
Shlomo Touboul)
) Art Unit: 2131
Application No: 09/539,667)
)
Filed: March 30, 2000)
)
For: SYSTEM AND METHOD FOR)
PROTECTING A COMPUTER)
AND A NETWORK FROM)
HOSTILE DOWNLOADABLES)
)

Assistant Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AND RESPONSE TO OFFICE ACTION UNDER 37 C.F.R. §1.111

Sir:

In response to the Office Action dated July 1, 2003 and pursuant to 37 C.F.R. §1.111, applicant respectfully requests that the above-identified application be amended as follows:

A

IN THE ABSTRACT OF THE DISCLOSURE:

Kindly replace the Abstract of the Disclosure with the following text:

a1

-- A computer-based method for generating a Downloadable ID to identify a Downloadable, including obtaining a Downloadable that includes one or more references to software components required by the Downloadable, fetching at least one software component identified by the one or more references, and performing a function on the Downloadable and the fetched software components to generate a Downloadable ID. A system and a computer-readable storage medium are also described and claimed. --

A

IN THE CLAIMS:

Kindly cancel claims 9 and 19 without prejudice.

Please substitute the following claims for the pending claims with the same number:

1. (Currently amended) A computer-based method for generating a Downloadable ID to identify a Downloadable, comprising [the steps of]:

obtaining a Downloadable that includes one or more references to software components required by the Downloadable;

fetching[, if the Downloadable includes one or more references to a component,] at least one software component identified by the one or more references; and

performing a function on the Downloadable and [all] the fetched software components [fetched] to generate a Downloadable ID.

2. (Original) The method of claim 1, wherein the Downloadable includes an applet.

A2

3. (Currently amended) The method of claim 1, wherein the Downloadable includes an [ActiveX^{TM}] active software control.

4. (Original) The method of claim 1, wherein the Downloadable includes a plugin.

5. (Original) The method of claim 1, wherein the Downloadable includes HTML code.

6. (Original) The method of claim 1, wherein the Downloadable includes an application program.

7. (Original) The method of claim 1, wherein the function includes a hashing function.

A

8. (Currently amended) The method of claim 1, wherein [the step of] said fetching includes [the step of] fetching the first software component referenced by the Downloadable.

9. (Cancelled)

10. (Currently amended) The method of claim 1, wherein [the step of] said fetching includes fetching all software components referenced by the Downloadable.

11. (Currently amended) A system for generating a Downloadable ID to identify a Downloadable, comprising:

a communications engine for obtaining a Downloadable that includes one or more references to software components required by the Downloadable; and

an ID generator coupled to the communications engine for fetching[, if the Downloadable includes one or more references to a component,] at least one software component identified by the one or more references, and for performing a function on the Downloadable and [all] the fetched software components [fetched] to generate a Downloadable ID.

12. (Original) The system of claim 11, wherein the Downloadable includes an applet.

13. (Currently amended) The system of claim 11, wherein the Downloadable includes an [ActiveX^{TM}] active software control.

14. (Original) The system of claim 11, wherein the Downloadable includes a plugin.

15. (Original) The system of claim 11, wherein the Downloadable includes HTML code.

A

16. (Original) The system of claim 11, wherein the Downloadable includes an application program.

17. (Original) The system of claim 11, wherein the function includes a hashing function.

18. (Currently amended) The system of claim 11, wherein the ID generator fetches the first software component referenced by the Downloadable.

19. (Cancelled)

20. (Currently amended) The method of claim 11, wherein the ID generator fetches all software components referenced by the Downloadable.

21. (Currently amended) A system for generating a Downloadable ID to identify a Downloadable, comprising:

means for obtaining a Downloadable that includes one or more references to software components required by the Downloadable;

means for fetching[, if the Downloadable includes one or more references to a component,] at least one software component identified by the one or more references; and

means for performing a function on the Downloadable and [all] the fetched software components [fetched] to generate a Downloadable ID.

22. (Currently amended) A computer-readable storage medium storing program code for causing a computer to perform the steps of:

obtaining a Downloadable that includes one or more references to software components required by the Downloadable;

fetching[, if the Downloadable includes one or more references to a component,] at least one software component identified by the one or more references; and

performing a function on the Downloadable and [all] the fetched software components [fetched] to generate a Downloadable ID.

A

REMARKS

Applicant has carefully studied the outstanding Office Action. The present amendment is intended to place the application in condition for allowance and is believed to overcome all of the objections and rejections made by the Examiner. Favorable reconsideration and allowance of the application are respectfully requested.

Applicant has canceled claims 9 and 19, and amended claims 1, 3, 8, 10, 11, 13, 18 and 20 - 22 to more properly claim the present invention. No new matter has been added. Claims 1 – 8, 10 – 18 and 20 – 22 are presented for examination.

Applicant notes that the page headers of the Office Action indicate an incorrect Application/Control Number.

In paragraphs 2 and 3 of the Office Action, the Examiner has objected to the abstract of the disclosure. Accordingly, applicant has amended the abstract so as to conform to the proper language and format.

In paragraphs 4 and 5 of the Office Action, the Examiner has rejected claims 1, 11, 21 and 22 under the judicially created doctrine of double patenting. Accordingly, applicant is submitting a terminal disclaimer with the present amendment.

In paragraphs 6 and 7 of the Office Action, the Examiner has rejected claims 3 and 13 under 35 U.S.C. §112, second paragraph as being indefinite. Applicant has amended these claims accordingly.

In paragraphs 8 and 9 of the Office Action, the Examiner has rejected claims 1, 7, 8, 10, 11, 17, 18, and 20 – 22 under 35 U.S.C. §102(e) as being anticipated by Apperson et al., U.S. Patent No. 5,978,484 ("Apperson").

In paragraphs 10 and 11 of the Office Action, the Examiner has rejected claims 2 – 4 and 12 – 14 under 35 U.S.C. §103(a) as being unpatentable over Apperson in view of Khare, "*Microsoft Authenticode Analyzed*", July 22, 1996, xent.com/FoRK-archive/summer96/0338.html, pg. 1 and 2 ("Khare").

In paragraph 12 of the Office Action, the Examiner has rejected claims 5, 6, 9, 15, 16 and 19 under 35 U.S.C. §103(a) as being unpatentable over Apperson. Applicant has canceled claims 9 and 19 without acquiescence to the Examiner's reasons for rejection and respectfully submits that rejection of those claims is thus rendered moot.

A

Distinctions between Claimed Invention and U.S. Patent No. 5,978,484 to Apperson et al in view of Khare, "Microsoft Authenticode Analyzed", July 22, 1996, xent.com/FoRK-archive/summer96/0338.html, pg. 1 and 2

The present invention concerns generation of an ID for mobile code downloaded to a client computer, referred to as a Downloadable. Specifically, the present invention fetches software components required by the Downloadable, and performs a hashing function on the Downloadable together with its fetched components (original specification / page 3, lines 11 – 14; page 15, lines 21 – 24; page 19, line 21 – page 20, line 6; FIG. 8). Thus, for a Java applet, the present invention fetches Java classes identified by the applet bytecode, and generates the Downloadable ID from the applet and the fetched Java classes; and for an ActiveX^{TM} control, the present invention fetches components listed in its .INF file, and generates a Downloadable ID from the ActiveX^{TM} control and the fetched components (original specification / page 9, lines 15 – 18).

An advantage of the present invention is that it produces the same ID for a Downloadable, regardless of which software components are included with the Downloadable and which software components are only referenced (original specification / page 9, lines 18 – 20; page 20, lines 5 and 6). The same Downloadable may be delivered with some required software components included and others missing, and in each case the generated Downloadable ID will be the same. Thus the same Downloadable is recognized through many equivalent guises.

Apperson describes use of digital certificates to authorize privileges for executable code, such as file I/O privileges, network privileges and registry privileges (Apperson / col. 2, lines 41 – 53; col. 4, lines 33 – 43; FIG. 2).

Khare describes Microsoft Corporation's implementation of digital signatures, referred to as Authenticode, as applied to ActiveX controls and Java applets.

In distinction to the present invention, Apperson and Khare do not teach fetching software components of executable code. In order to further clarify this distinction, applicant has amended the claims so as to refer to software components required by the Downloadable.

In paragraph 9 of the Office Action, the Examiner has indicated that Apperson discloses fetching components of a Downloadable. Applicant respectfully submits that Apperson's privilege request code does not include components of a Downloadable, but instead includes a list of "*privileges or*

A

privilege categories that the executable code might perform on the client machine" (Apperson / col. 2, lines 45 – 47).

The rejections of claims 1 –8 and 10 in paragraphs 8 - 12 of the Office Action will now be dealt with specifically.

As to amended independent method claim 1, applicant respectfully submits that the limitation in claim 1 of:

"*fetching at least one software component identified by the one or more references*"

is neither shown nor suggested in Apperson or Khare.

Because claims 2 – 8 and 10 depend from claim 1 and include additional features, applicant respectfully submits that claims 2 - 8 and 10 are not anticipated or rendered obvious by Apperson and Khare, taken alone or in combination.

Accordingly claims 1 – 8 and 10 are deemed to be allowable.

As to amended independent system claim 11, applicant respectfully submits that the limitation in claim 11 of:

"*an ID generator coupled to the communications engine for fetching at least one software component identified by the one or more references*"

is neither shown nor suggested in Apperson or Khare.

Because claims 12 – 18 and 20 depend from claim 11 and include additional features, applicant respectfully submits that claims 12 – 18 and 20 are not anticipated or rendered obvious by Apperson and Khare, taken alone or in combination.

Accordingly claims 12 – 18 and 20 are deemed to be allowable.

As to amended independent system claim 21, applicant respectfully submits that the limitation in claim 21 of:

"*means for fetching at least one software component identified by the one or more references*"

is neither shown nor suggested in Apperson or Khare.

Accordingly claim 21 is deemed to be allowable.

As to amended independent system claim 22, applicant respectfully submits that the limitation in claim 22 of:

"*fetching at least one software component identified by the one or more references*"

is neither shown nor suggested in Apperson or Khare.

Accordingly claim 22 is deemed to be allowable.

A

Support for Amended Claims in Original Specification

Regarding amended claims 1, 8, 10, 11, 18 and 20 - 22, fetching software components is described in the original specification on page 9, lines 13 – 18 and FIG. 8.

For the foregoing reasons, applicant respectfully submits that the applicable objections and rejections have been overcome and that the claims are in condition for allowance.

If the Examiner has any questions or needs any additional information, the Examiner is invited to telephone the undersigned attorney at (650) 843-3392. If for any reason an insufficient fee has been paid, please charge the insufficiency to Deposit Account No. 05-0150.

Date: July 31, 2003

Respectfully submitted,

Squire, Sanders & Dempsey L.L.P.
600 Hansen Way
Palo Alto, CA 94304-1043
Telephone: (650) 856-6500
Facsimile: (650) 843-8777

By:
Marc A. Sockol
Attorney for Applicant
Registration No. 40,823

A

FORRESTER & BOEHMERT

Professional Representation at the European Patent Office* Zugelassen zur Vertretung vor dem Europäischen Patentamt*

Forrester & Boehmert, Pettenkoferstrasse 20-22, D-80336 München
European Patent Office
Erhardtstrasse 27

D-80469 MÜNCHEN

PETTENKOFERSTRASSE 20-22
D-80336 MÜNCHEN
GERMANY
Telephone +49 (0) 89 55 96 80
Fax +49 (0) 89 34 70 10
E-Mail fbmunich@forresterboehmert.com

Our ref
Unser Zeichen
FB8354
E13357EP - SJP/JCC/lp

Your ref
Ihr Zeichen

Date
Datum
21 December 2005

Dear Sirs

Re: European Patent Application No. 97950351.3
Finjan Software, Ltd
<u>Our Ref: FB8354/ E13357EP</u>

With reference to the official letter of 15 June, we are filing herewith amended Claims 1 to 60 to replace the claims at present on file. We also file herewith amended pages 1 and 1A to replace page 1 of the description at present on file and an amended page 18 of the description to replace page 18 at present on file.

The amended set of claims has only two independent claims, Claim 1 to the method and Claim 29 to the system. The independent Claims 1 and 29 in the amended set have been cast in the two-part form in accordance with Rule 29(1) EPC. The amended claims incorporate reference numerals from the drawings.

Amended Claims 2 to 10 correspond to original Claims 2 to 10 with clarifying amendment to Claims 2, 3 and 4; amended Claims 15 to 21 correspond to original Claims 15 to 21 with clarifying amendment; amended Claim 22 corresponds to original Claim 22 but with some amendment; amended Claim 23 corresponds to original Claim 26; amended Claims 24 and 25 correspond to original Claims 24 and 25 with amendment for clarity; amended Claim 26 is based on original Claim 29 whilst amended Claim 27 corresponds to original Claim 30. Claim 28 is new and basis for this claim is discussed below. Claim 29 in the amended set of claims is based on original Claim 31 but written to include the limitations of the new Claim 1. Claims 30 to 57 in the amended set correspond to original Claims 32 to 59; new Claims 58 and 59 correspond to original Claims 61 and 62 and Claim 60 in the amended set of claims filed herewith is new. Basis for new Claim 60 is discussed below.

Website: www.forresterboehmert.com

Accounts Munich/Konten München: HypoVereinsbank AG (BLZ 700 202 70) 577870 Postbank München (BLZ 700 100 80) 135-808.
Associated Firms/Assoziiert mit: Forrester Ketley & Co. London - Birmingham VAT No. DE 114441475
Boehmert & Boehmert München - Bremen - Berlin - Düsseldorf - Frankfurt - Bielefeld – Potsdam - Kiel - Paderborn - Landshut - Höhenkirchen - Alicante - Paris

JA2063

As regards new Claim 1, this now recites a method of operating an internal network security system (*cf* original page 3, lines 1 and 2; page 4, line 15 etc.) coupling (*cf* page 2, lines 15 and 18) at least one client computer (*cf* page 2, lines 13 to 15 and original Claims 11, 12, 13 and 14), with an external network, the method comprising receiving, by said internal network security system, from said external network, data addressed to the client computer (the reference to "data addressed to the client computer" has been inserted simply in view of the fact that the claim has now been written in the requisite two-part form and the applicants are anxious not to attribute more to the prior art than is justified. Furthermore, it is submitted that it is clear from the preamble, the context of the specification as a whole, and the description of the preferred embodiment, that the purpose of the internal network security system such as illustrated schematically at 110 in Figure 1, is to protect individual client computers or an internal computer network (such as what is now commonly referred to a LAN or local area network) from some elements of what may be received from the Internet, (or other "external computer network" in the language of page 4, line 14), whilst allowing non-hostile elements of what may be received to pass to such client computers. Whilst the original description may not have referred to what may be received from the Internet or other external computer network as data, the man in the art at the priority date of the application would have been fully aware that what is transmitted around the Internet and between computers is essentially data, albeit generally embodied in electrical or electromagnetic signals of various kinds. Likewise it would have been routine knowledge for the man in the art that the operation of the Internet and of computer networks involves the addressing of data, e.g. in the in the form of data packages, to individual computers and that the references in the present application as originally filed to Downloadables being addressed to particular clients necessarily involved such Downloadables being incorporated in data or data packages addressed to particular client computers and that the system of the present invention, like those of the prior art, must receive everything addressed to any of the client computers which is its function to protect, if it is to receive or examine Downloadables in particular.

The characterising part of new Claim 1 indicates that the method includes examining the Downloadables, (defined as executable application programs). It is submitted that the source of Downloadable is of no relevance to the method, beyond the fact that they arrive from the external network and clearly they will only run on the destination computer if the internal network security system allows them to pass to the client computer. The amended claim indicates that the Downloadables are examined according to a security policy defined by at least one test (see page 2, line 19 and page 3, lines 1 and 2). (Original Claim 1 referred to comparing Downloadables <u>against</u> a security policy, but is submitted that it is clear what was meant). It is submitted that it is clear from pages 2 and 3 of the description, and from the description of the preferred embodiment e.g. page 14, lines 16 and 21; page 15, lines 5, 7 and 8 and page 15, lines 17 to 22; page 16, line 15 etc. and from the context of the specification as a whole, that each security policy referred to in the specification originally filed is simply a test or a set of tests which determines whether a Downloadable is to be passed to a particular client computer to which it is addressed or discarded. It is submitted that the wording "examining... Downloadables according to a security policy defined by at least one text" does no more than convey succinctly what was clearly disclosed to the man in the art by the specification as originally filed. Amended Claim 1 now includes the limitation of the method conducting the test (or tests) concerned with reference to a DSP comprising a list of suspicious computer operations that the received Downloadable may attempt. Original Claim 2 required the method to include the step of comparing a DSP against a security policy. Page 7, lines 10 to 12 and page 9, lines 16 to 19 of the description indicate that a DSP includes <u>the</u> list of all potentially hostile or suspicious computer operations that may be attempted by a specific Downloadable. It will be noted at page 10, lines 11 to page 11, line 2 also make it clear that a DSP is specific to a particular Downloadable and this is, of course, inherent in original Claim 2. It is submitted that it would have been clear to the man in the art, e.g. from the reference to a "Downloadable security profile" in original Claim 2 and on page 2, that the definition of a DSP on page 7 and page 9 referred to above was not intended simply to be a definition confined to the particular embodiment under discussion in these pages but was intended to be a definition of a DSP generally as that term was used in the specification. The description of Path 2 at page 9, lines 11 to 16 makes it clear that if a DSP is not already known by the system for a particular Downloadable, the system generates one, (as recited in original Claim 2). The remainder of new Claim 1 is effectively a repetition of the last three lines of original Claim 1 augmented, by way of clarification, to indicate what is doing what, with a view to addressing the points in item 3.2 of the official letter.

New Claim 11 is based on original Claim 11 but explicitly states what is implicit in original Claim 1, that the internal network security system is interposed between the external network and an internal network of client computers and can apply any of a plurality of security policies. It is submitted that it is inherent in original Claims 11, 12 and 13 that the system may have a number of security policies at its disposal, a fact which is also made clear in the description from original page 6 onwards. New Claim 12 is dependent on new Claim 11 and is of the same scope as original Claim 11. New Claim 13 is based on original Claim 13 and original Claim 22.

New Claim 26 has basis in original Claim 29 and original Claim 54 and also at page 6, line 14; page 7, line 5 and page 10, lines 22 and 23. Original page 2, lines 19 to 22 indicate that the security policy may indicate tests including a comparison of the Downloadable security profile against access control lists. Page 7, lines 4 to 6 indicate that "the security policy 305 may include ... access control lists" and page 10, lines 22 to 23 indicate that the access control list contains criteria indicating whether to pass or fail the Downloadable, (see also page 15, lines 2 to 5). Again, page 2, lines 21 and 22 referred to comparison of the Downloadable security profile against the access control lists, (see also original page 10, lines 22 and 23).

New Claims 28 and 60 (new Claim 60 is the "system" equivalent to new Claim 28), find basis in original Claims 16, 18, 39, 41, 65 and 66 and at page 2, lines 15 to 19; page 7, line 15 to page 9, line 9; page 9, lines 11 to 23; and also page 10, lines 11 to 18.

New dependent method claim 28 includes the limitation of checking an incoming Downloadable to determine if it is already known to the system, and obtaining its Downloadable security profile (DSP) from memory, if it is known. This is described in the original specification at page 14. lines 22 - 24, and at page 17, line 24 - page 18, line 1. New dependent system claim 60 is intended to parallel new claim 28.

Amended Claim 29, as previously noted, is effectively the previous Claim 31 with the limitations of new Claim 1 added. Basis for the reference to the "logical engine" appears in original Claim 60 and 61.

Whilst the description as originally filed does not appear to refer to the internal network security system (110) as being specifically a computer system, it is clear from the description as a whole that this is what it is and in the light of the last sentence of item 3.3 of the official letter, the internal network security system is referred to as a computer system in Claims 1 and 29. The applicants did contemplate referring to the system (110) in the amended claims as a server computer or a gateway server, but refrained from doing so in the absence of any specific reference to the terms "server" or "gateway" in the specification as originally filed.

As noted above, the amended claims now incorporate a definition of the term "Downloadable" as being an executable application program. Furthermore, the term "security policy" has now been clarified as being defined by at least one test for determining whether or not to allow a Downloadable or discarded. The claims have also been amended to clarify that a "client" is a "client computer".

As regards document D2 referred to by the Examiner, the Applicant believes that the date of document D2 is erroneously listed as March 1988 in the Supplemental European Search Report. It is believed that the true date of D2 is March 1998, as is clear from the relevant web page (at http://www.isoc.org/isoc/conferences/ndss) of the Internet Society, and, indeed, from the dates given for the references cited in D2 itself.

In Paragraph 5 of the Office Action, the Examiner has indicated that the matter of claims 13,14, 20,22, 23, 26, 38,47, 52 and 53 is not disclosed in any of the documents cited in the search report, and that these claims would be allowable if re-drafted in independent form. We thank the Examiner for this indication, but the applicants consider that the independent claims in the amended set filed herewith cover a more important aspect of the invention.

In Paragraph 8 the Examiner has requested that documents D1 and D2 be identified in the description and that the relevant background art disclosed therein be briefly discussed. Regarding document D1, the amended pages 1 and 1a identify and briefly discuss this. Regarding document D2, as noted above, it is submitted that the correct date for this document is March 1998, and not March 1988 as indicated in the Supplemental European Search Report, so that document D2 is not prior art. Accordingly, no reference to D2 has been inserted.

Page 18 of the description has been amended to address paragraph 9.2 of the Office Action.

We enclose a copy of the original claims in which we have marked, as far as practicable, the amendments made.

As regards the cited prior art, D1, the Examiner is asked to have regard to the following:-

Distinctions between the claimed Invention and U.S. Patent No. 5,412,717 to Fisher (Document D1).

Document D1 describes a method and apparatus for controlling the behaviour of programs that run locally on a computer. Through use of program authorisation information (PAI) an administrator or user can prescribe operations that a program is permitted to perform or forbidden from performing. (D1 / col 2, lines 34 - 36). When the program is run, its corresponding PAI is enforced in real-time by quarantining the program to run in an "isolation" mode within a "safety box." Thus, when the program attempts to perform an operation to access system resources such as files, or attempts to invoke another program, or attempts to open a communication socket, or attempts to send e-mail, or attempts to solicit a user signature, or such other operation that can damage or compromise the security of a computer or its data, the operation is routed through a "supervisor" function which examines the PAI and blocks forbidden operations on-the-fly. (D1 / col. 2, lines 36 - 43; col. 18, line 57 - line 19, line 20, line 5).

In distinction to document D1, the present invention enforces security at a gateway server computer, which is remote from the client computer at which a suspicious program is intended to run. The gateway server computer receives a Downloadable intended for the client computer, and pre-scans the Downloadable to determine suspicious operations that the Downloadable may try to execute. If the Downloadable passes a security check, based on a security policy, then the Downloadable is forwarded to the client for normal execution. Otherwise, the Downloadable is preferably blocked from the client, or modified prior to forwarding to the client.

It may thus be appreciated that whereas document D1 teaches real-time control of malicious programs running on a client computer, by running such programs within a "safety box," the present invention pre-analyses such programs at a gateway computer, prior to forwarding them to the client computer for which they are intended. The present invention cannot use real-time "safety box" intervention as in document D1, since the present invention must make safety determination at a gateway computer that is remote from the client computer. As described at page 4, lines 21-23 of the present specification, the present invention protects the client computer from receiving dangerous programs. Document D1, on the other hand, describes protecting dangerous programs that are running on the client computer from malicious behaviour. The amended independent claims 1 and 29 are intended to clarify these distinctions.

It is believed that the amendments effected hereby should at least form a basis for allowance of the patent application, but we hereby request, simply as a precaution, Oral Proceedings in the event that the Examiner is minded to reject the application in the next official action. We would, however, prefer to discuss any further points with the Examiner by telephone, if the Examiner would consider that helpful or convenient.

Yours faithfully

FORRESTER & BOEHMERT

P.S. Please acknowledge safe receipt hereof by returning the enclosed copy letter duly endorsed.

IBM Dictionary of Computing

- ▼ The most comprehensive computing dictionary ever published
- ▼ More than 18,000 entries

IBM DICTIONARY OF COMPUTING

Compiled and edited by

GEORGE McDANIEL

McGRAW-HILL, INC.
New York San Francisco Washington, D.C. Auckland Bogotá
Caracas Lisbon London Madrid Mexico City Milan
Montreal New Delhi San Juan Singapore
Sydney Tokyo Toronto

Limitation of Liability

While the Editor and Publisher of this book have made reasonable efforts to ensure the accuracy and timeliness of the information contained herein, neither the Editor nor the Publisher shall have any liability with respect to loss or damage caused or alleged to be caused by reliance on any information contained herein.

6 7 8 9 0 DOC/DOC 9 9 8 7

ISBN 0-07-031488-8 (HC)
ISBN 0-07-031489-6 (PBK)

The sponsoring editor for this book was Daniel A. Gonneau and the production supervisor was Thomas G. Kowalczyk.

Printed and bound by R. R. Donnelley & Sons Company.

Tenth Edition (August 1993)

This is a major revision of the *IBM Dictionary of Computing,* SC20-1699-8, which is made obsolete by this edition. Changes are made periodically to the information provided herein.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country. Comments may be addressed to IBM Corporation, Department E37/656, P. O. Box 12195, Research Triangle Park, NC 27709.

International Edition

When ordering this title, use ISBN 0-07-113383-6.

This book is printed on acid-free paper.

[illegible] yield the sum and the difference at the same [illegible]

add file A file to which records are being added.

addition See parallel addition, serial addition.

addition without carry Deprecated term for nonequivalence operation.

additive attribute (1) In PL/I, a file description characteristic that must be stated explicitly or implied by another explicitly stated characteristic. Contrast with alternative attribute. (2) In data communications, a collection of multipoint addresses. Each address can be associated with an individual communications session. Contrast with alternative attribute.

additive color system In computer graphics, a system that reproduces an image by mixing (adding) appropriate quantities of red, green, and blue light. See also primary color. Contrast with subtractive color system.

add mode In addition and subtraction operations, a mode in which the decimal marker is placed at a predetermined location with respect to the last digit entered. (I) (A)

add operation (1) A disk or diskette operation that adds records to an existing file. (2) An operation caused by an add instruction.

address (1) A character or group of characters that identifies a register, a particular part of storage, or some other data source or destination. (A) (2) To refer to a device or an item of data by its address. (I) (A) (3) In word processing, the location, identified by an address code, of a specific section of the recording medium or storage. (T) (4) A name, label, or number identifying a location in storage, a device in a system or network, or any other data source. (5) In data communication, the unique code assigned to each device or workstation connected to a network.

addressability (1) In computer graphics, the number of addressable points on a display surface or in storage. (2) In micrographics, the number of addressable points, within a specified film frame, computed as follows: the number of addressable horizontal points by the number of addressable vertical points; for example, 4000 by 4000. (A) (3) The ability to locate an item in online storage.

addressability measure On a display screen, the number of addressable points within the display space.

addressable horizontal positions (1) In micrographics, the number of positions, within a specified film frame, at which a vertical line can be placed. (A) (2) In computer graphics, the number of positions on a display surface at which a full-length display column can be placed.

addressable point In computer graphics, any point of a device that can be addressed. (I) (A) Synonymous with addressable position. See Figure 4.

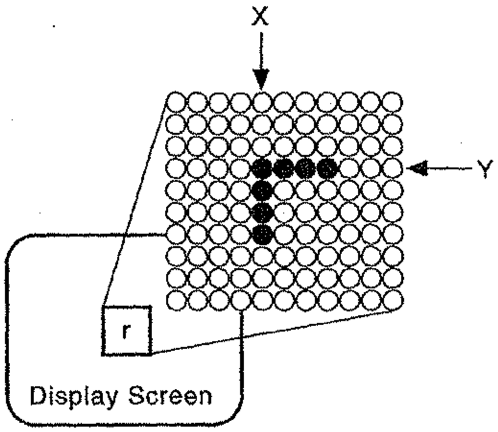


Figure 4. Addressable Point

addressable position Synonym for addressable point.

addressable vertical positions (1) In micrographics, the number of positions, within a specified film frame, at which a horizontal line can be placed. (A) (2) In computer graphics, the number of positions on a display surface at which a full-length display column can be placed. (A)

address administration The assignment of LAN individual addresses locally or on a universal basis. (T)

address aliasing See network address translation.

address base The field of an address control vector that designates the origin of a logical address space in the processor address space. It is concatenated with a logical address during dynamic address relocation.

address book A collection of entries containing information that can be referenced and used to perform OS/2 Office functions. In the OS/2 Office product, there is a Personal address book and a Public address book contained inside each directory.

address book entry An entry of information contained inside a Public or Personal address book for use performing OS/2 Office functions.

records or keys. Usually, it precedes a merge phase in which the sequences created are reduced to one by an external merge. (4) In VSAM, when building an alternate index, the sorting of the alternate keys into ascending sequence by using virtual storage obtained through a GETVIS. See also external sort.

internal storage (1) Storage that is accessible by a processor without the use of input/output channels. Internal storage may include other kinds of storage such as cache memory and registers. Synonymous with internal memory. (T) (2) Deprecated term for main storage. (3) Synonym for processor storage.

Notes:

1. *Internal storage usually refers to one or more storage devices that together provide the total program-addressable execution space of main storage.*
2. *Internal storage includes processor storage and may include other kinds of storage accessed by a processor, such as cache storage and special registers.*

internal trace table Synonym for CP trace table.

internal writer A facility in the job entry subsystem (JES2 or JES3) that allows user-written output writers to write data on devices not directly supported by the job control manager.

international standard A standards document that is given final approval by the International Organization for Standardization.

International Telecommunication Union (ITU) The specialized telecommunication agency of the United Nations, established to provide standardized communication procedures and practices, including frequency allocation and radio regulations worldwide.

internet A collection of packet-switching networks that are physically interconnected by Internet Protocol (IP) gateways. These networks use protocols that allow them to function as a large, composite network.

Internet A wide area network connecting thousands of disparate networks in industry, education, government, and research. The Internet network uses TCP/IP as the standard for transmitting information.

Internet address The numbering system used in TCP/IP Internetwork communications to specify a particular network or a particular host on that network with which to communicate. Internet addresses are commonly denoted in dotted decimal form.

Internet Control Message Protocol (ICMP) A protocol used by a gateway to communicate with a source host, for example, to report an error in a datagram. It is an integral part of the Internet Protocol (IP).

Internet Protocol (IP) A protocol used to route data from its source to its destination in an Internet environment.

Internet router A device that enables an Internet Protocol host to act as a gateway for routing data between separate networks that use a specific adapter.

internetwork Any wide area network connecting more than one network.

internetworking Communication between two or more networks.

internodal awareness In an ACF/TCAM extended networking, a function used by TCAM systems to share information. This information includes the status of TCAM systems, the status of application programs in TCAM systems, and the contents of selected key-table entries. This function is provided by node path system service programs in the various TCAM systems that communicate with each other.

internodal destination queue In an ACF/TCAM extended networking, a destination queue for an external logical unit (LU) that is a partner in a utility session.

internodal message handler (IMH) In an ACF/TCAM extended networking, a message handler that processes messages flowing on utility sessions.

internodal sequence number synchronization In an ACF/TCAM extended networking, the function of a particular system service program that operates in conjunction with the internodal message handler. Internodal sequence number synchronization is used to request retransmission from any TCAM node of sequence-numbered messages not received in a utility session and to retransmit sequence-numbered messages flowing in a utility session when requested to do so by another TCAM node or an extended operator command.

internodal sequence prefix In an ACF/TCAM extended network, a control block that contains sequence-number information for messages flowing in utility sessions.

internode routing The capability of path control to route PIUs from half-sessions to data link control and from data link control to half-sessions for sessions between NAUs that reside in different nodes.

intern procedure In the AIX operating system, the procedure of defining an atom.

Dictionary of Computer and Internet Terms

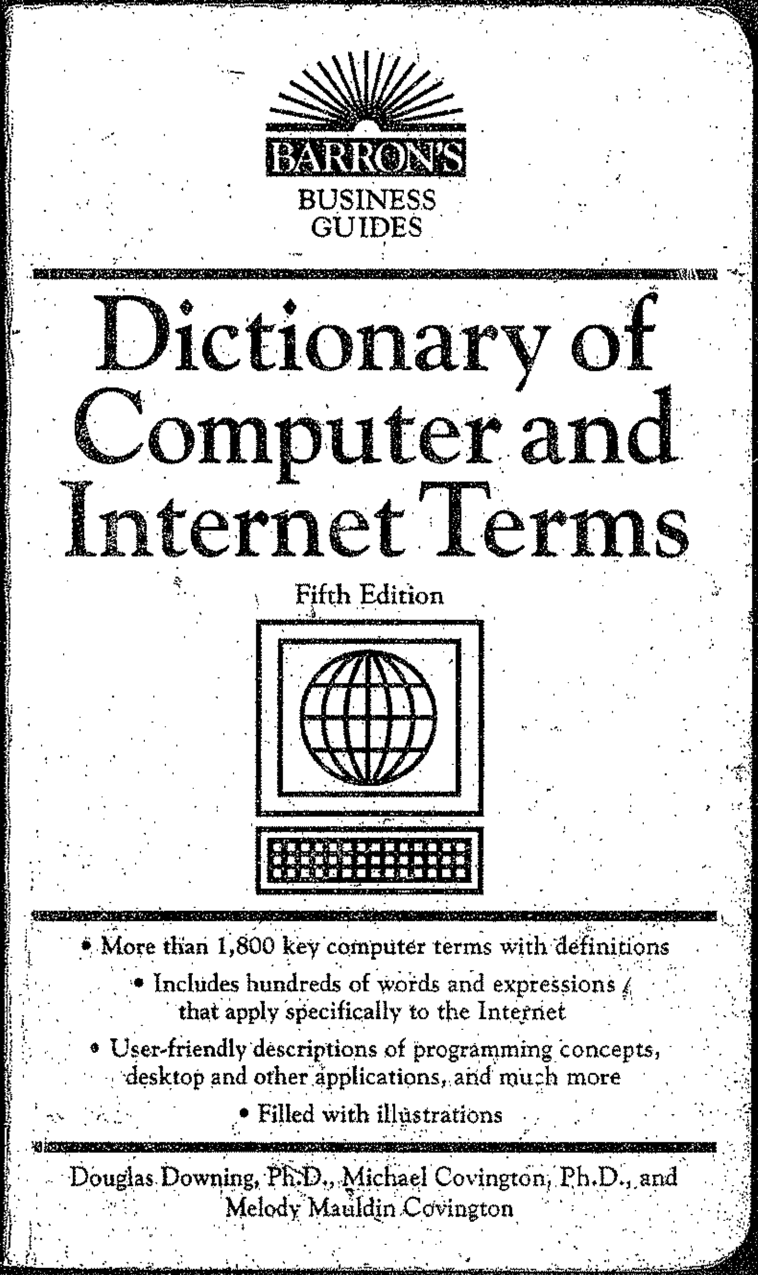
Fifth Edition

Douglas A. Dowining, Ph.D.
School of Business and Economics
Seattle Pacific University

Michael A. Covington, Ph.D.
Artificial Intelligence Center
The University of Georgia

Melody Mauldin Covington
Graphic Designer
Athens, Georgia

BARRON'S



BARRON'S
BUSINESS
GUIDES

Dictionary of Computer and Internet Terms

Fifth Edition

- More than 1,800 key computer terms with definitions
- Includes hundreds of words and expressions that apply specifically to the Internet
- User-friendly descriptions of programming concepts, desktop and other applications, and much more
- Filled with illustrations

Douglas Downing, Ph.D., Michael Covington, Ph.D., and
Melody Mauldin Covington

All inquiries should be addressed to:
Barron's Educational Series, Inc.
250 Wireless Boulevard
Hauppauge, New York 11788

Library of Congress Catalog Card No. 96-9250

International Standard Book No. 0-8120-9811-0

Library of Congress Cataloging-in-Publication Data

Downing, Douglas.
Dictionary of computer terms / Douglas A. Downing, Michael Covington, Melody Mauldin Covington—5th ed.
p. cm.
Previous eds. published under title: Dictionary of computer terms.
ISBN 0-8120-9811-0
1. Computers—Dictionaries. 2. Internet (Computer network)—Dictionaries. I. Covington, Michael A., 1957– . II. Covington, Melody Mauldin. III. Downing, Douglas. Dictionary of computer terms. IV. Title.
QA76.15.D667 1996
004'.03—dc20 96-9250
CIP

PRINTED IN THE UNITED STATES OF AMERICA

98765432

ABOUT THE AUTHORS

Douglas Downing teaches economics at Seattle Pacific University. He is the author of several books in Barron's Easy Way series including *Computer Programming in BASIC*, *Computer Programming in Pascal* (with Mark Yoshimi), *Algebra*, *Trigonometry*, *Calculus*, and *Statistics* (with Jeff Clark). He is also the author of *Business Statistics* and *Quantitative Methods* (both with Jeff Clark for the Barron's Business Review series), *Computers and Business Tasks*, and *Dictionary of Mathematics Terms*, all published by Barron's Educational Series, Inc. He holds the Ph.D. degree in economics from Yale University.

Michael Covington is a research scientist and teacher in natural language processing and artificial intelligence at the University of Georgia. He is a contributing editor for *PC Techniques* and *Electronics Now* magazines and is the author of *Computer Science Study Keys* (published by Barron's), *Prolog Programming in Depth* (with Donald Nute), *Astrophotography for the Amateur*, *Syntactic Theory in the High Middle Ages*, and numerous articles in scholarly journals and computer and electronics magazines. He holds the Ph.D. degree in linguistics from Yale University.

Melody Mauldin Covington is a graphic designer living in Athens, Georgia. She is author of *Dictionary of Desktop Publishing* (published by Barron's) and was formerly art director of *The Marietta Daily Journal* and *Neighbor Newspapers* in suburban Atlanta.

INTRANET the opposite of INTERNET; a network confined to a single organization (but not necessarily a single site).

INVERT

1. to turn an image into a photographic negative of itself, substituting black for white and white for black, and changing colors to their complements.
2. (less commonly) to turn an image upside down.

For other senses see INVERTER.

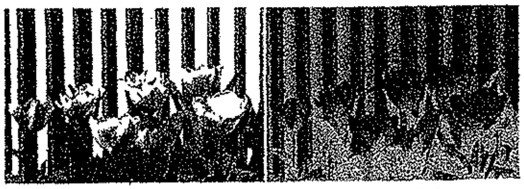


FIGURE 115. INVERTED PHOTOGRAPH

INVERTER

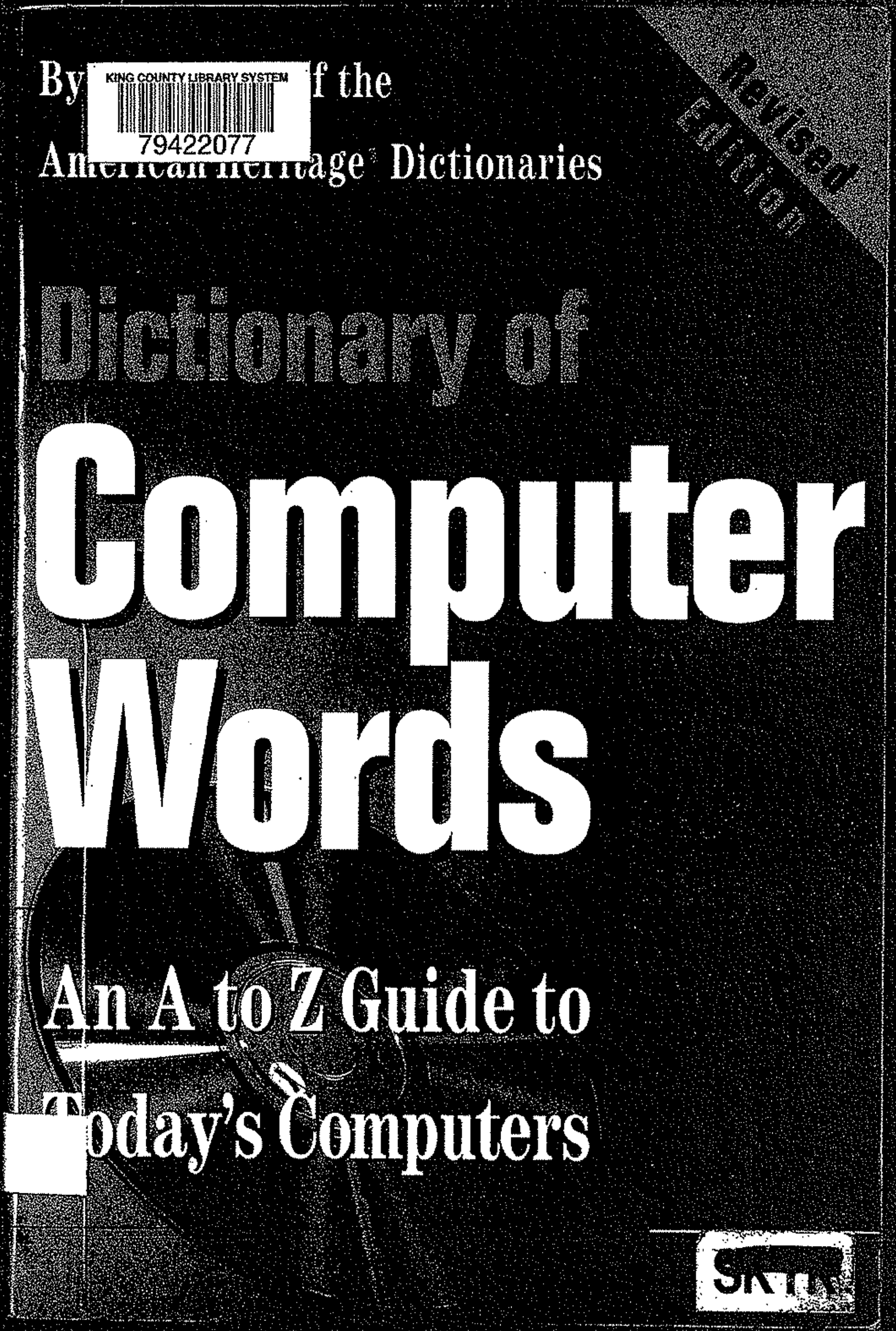
1. a NOT gate. (*See* NOT GATE.)
2. a device that converts direct current to alternating current for power supply purposes (e.g., to power a computer from a car battery).

IP ADDRESS (Internet Protocol address) the numeric address of a machine, in the format used on the Internet. For example, the IP address of one of the University of Georgia's computers is 128.192.12.9. *Contrast* DOMAIN ADDRESS. *See* INTERNET.

IP SPOOFING *see* SPOOFING.

IRC (Internet Relay Chat) a multi-user conversation conducted over the Internet in real time. Fig. 116 shows what a chat session looks like. Numerous CHANNELS (conversation forums) exist. Participants normally identify themselves by nicknames.

In addition to typing remarks for transmission to the other participants, the IRC user can type commands such as `/list` to see what channels are available, `/join #frogs` to join a channel called `frogs` or create it if it doesn't exist, and `/bye` to sign off.



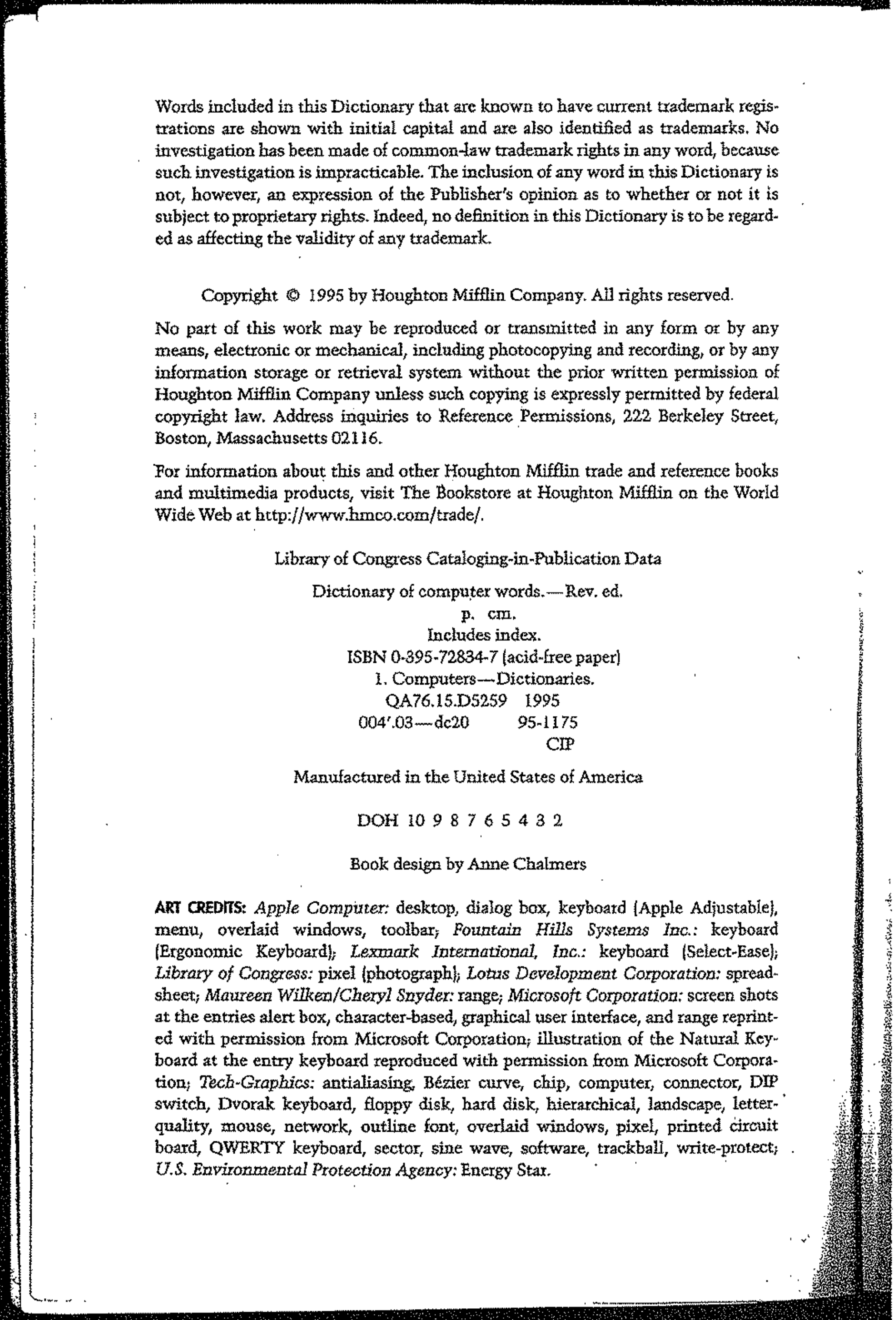
By f the

American Heritage Dictionaries

Revised Edition

Dictionary of Computer Words

An A to Z Guide to Today's Computers



For information about this and other Houghton Mifflin trade and reference books and multimedia products, visit The Bookstore at Houghton Mifflin on the World Wide Web at http://www.hmco.com/trade/.

Library of Congress Cataloging-in-Publication Data

Dictionary of computer words.—Rev. ed.
p. cm.
Includes index.
ISBN 0-395-72834-7 (acid-free paper)
1. Computers—Dictionaries.
QA76.15.D5259 1995
004'.03—dc20 95-1175
CIP

Manufactured in the United States of America

DOH 10 9 8 7 6 5 4 3 2

Book design by Anne Chalmers

ART CREDITS: *Apple Computer:* desktop, dialog box, keyboard (Apple Adjustable), menu, overlaid windows, toolbar; *Fountain Hills Systems Inc.:* keyboard (Ergonomic Keyboard); *Lexmark International, Inc.:* keyboard (Select-Ease); *Library of Congress:* pixel (photograph); *Lotus Development Corporation:* spreadsheet; *Maureen Wilken/Cheryl Snyder:* range; *Microsoft Corporation:* screen shots at the entries alert box, character-based, graphical user interface, and range reprinted with permission from Microsoft Corporation; illustration of the Natural Keyboard at the entry keyboard reproduced with permission from Microsoft Corporation; *Tech-Graphics:* antialiasing, Bézier curve, chip, computer, connector, DIP switch, Dvorak keyboard, floppy disk, hard disk, hierarchical, landscape, letter-quality, mouse, network, outline font, overlaid windows, pixel, printed circuit board, QWERTY keyboard, sector, sine wave, software, trackball, write-protect; *U.S. Environmental Protection Agency:* Energy Star.

instruction is evaluated. In a compiled language that translation process happens only once, producing an object program that requires no further translation while it's running. See also *compiler*.

interrupt A signal to a *program* that demands an immediate response. The program may stop temporarily so that another action can be performed. It may also decide to ignore the interrupt. If it stops running, it saves its current work before executing whatever *instructions* are interrupting it. As soon as this is over, the program resumes. Interrupts may come from the *hardware* or the *software*. See also *exception*.

inverse video See **reverse video.**

I/O Abbreviation of **input/output.** Designating a *program* or *device*, such as a *mouse* or *printer*, that is used to *input* or *output data* rather than to process it. A *modem*, for example, both inputs and outputs data but does not process it.

IP address Abbreviation of **Internet protocol address.** The unique numerical sequence that serves as an identifier for an Internet server. An IP address appears as a series of numbers separated by dots or periods, as "155.44.208.3".

IRC Abbreviation of **Internet Relay Chat.** A network of Internet servers worldwide through which individual users can hold real-time on-line conversations. In an IRC, users can "talk" to each other as part of a group discussion on any of a number of specified topics.

IRMA board [UR-muh] An *expansion board* for *Macintosh* computers and IBM PC and compatible computers that lets them *emulate terminals*. This means that if you have an IRMA board, you can connect your computer to a *mainframe* for certain tasks while still using it as a *personal computer* for others.

ISA Abbreviation of **Industry Standard Architecture.** The *bus* architecture used for IBM PC/XT computers and IBM PC/AT computers. The ISA bus was originally created for the IBM

H
Only the Westlaw citation is currently available.

United States District Court,
E.D. Texas,
Marshall Division.
EPICREALM, LICENSING, LLC
v.
AUTOFLEX LEASING, INC., et al.
Epicrealm, Licensing, LLC
v.
Franklin Covey Co., et al.
Nos. 2:05CV163, 2:05CV356.

Oct. 30, 2006.

ORDER ADOPTING REPORT AND RECOMMENDATION REGARDING CLAIM CONSTRUCTION

DAVID FOLSOM, District Judge.

*1 The above-entitled and numbered civil action was heretofore referred to United States Magistrate Judge Caroline M. Craven pursuant to 28 U.S.C. § 636. The Report and Recommendation Regarding Claim Construction of the Magistrate Judge which contains her proposed findings of fact and recommendations for the disposition of such action has been presented for consideration. Plaintiff and Defendants both filed objections to the Report and Recommendation of the Magistrate Judge. The Court conducted a *de novo* review.

The Court, having reviewed the relevant briefing, finds the parties' objections are without merit. In their briefing, the parties address, among other things, the word "mechanism" as used in the Magistrate Judge's discussion of the construction of "Web page." *See* Report and Recommendation at pgs. 8- 9 ("As described within the specification, a Web page is a *mechanism* through which static and dynamic content may be displayed.")(emphasis added). A Web page may include static or dynamic content. However, the Magistrate Judge's construction of "Web page" as "Web content displayable through a Web browser" does not include the word "mechanism." The Court adopts the construction of "web page" as proposed by the Magistrate Judge, and with the exception of the use of the word "mechanism," the Court also adopts the reasoning of the Magistrate Judge with respect to the construction of "Web page."

The Court is of the opinion that the findings and conclusions of the Magistrate Judge are correct. Therefore, the Court hereby adopts the Report of the United States Magistrate Judge as the findings and conclusions of this Court.

IT IS SO ORDERED.

CAROLINE M. CRAVEN, Magistrate Judge.

REPORT AND RECOMMENDATION REGARDING CLAIM CONSTRUCTION

Pursuant to the provisions of 28 U.S.C. § 636(b)(1) and (3) and the Amended Order for the Adoption of Local Rules for Assignment of Duties to United States Magistrate Judges, the above-entitled and numbered cause of action was referred to the undersigned for pretrial purposes. Claim construction arguments in cause numbers 2:05-CV-163 and 2:05-CV-356 were combined, and Defendants submitted joint briefing. The Court conducted a claim construction hearing on July 13, 2006. This Report and Recommendation construes certain terms in United States Patent Nos. 5,894,554 ("the '554 Patent) and 6,415,335 (the '335 Patent).

I. BACKGROUND

The '554 Patent issued on April 13, 1999. The ' 335 Patent issued on July 2, 2002 and is a divisional application of the ' 554 Patent. The '554 Patent and the '335 Patent share a common specification. [FN1] The patents generally relate to managing Web sites. More particularly, the patents relate to managing dynamic Web page generation. Col. 2:15-23. The patents distinguish some Web pages as having a static nature that remains static until manually modified and other Web pages as being dynamic Web pages which contain content that is generated dynamically by retrieving the necessary requested data and generating the requested Web page dynamically. Col. 1:38-55. The patents describe prior art Web servers as handling both static and dynamic Web page requests. Col. 3:64-Col. 4:37; Figures 2-3. The techniques described in the patents include routing a Web request from the Web server to a Page server. The Page server may than process the request, and the Web server is released to process other requests. Col. 2:20-35; Col. 4:54-Col. 6:32. In this manner, dynamic Web pages may be generated by the Page servers.

FN1. References to the specification will refer to the column and line numbers of the '554 Patent.

*2 Some of the claim construction disagreements involve common themes. For example, in general the Plaintiff construes various terms so that Web servers and Page servers do not have to be separate machines while the Defendants seek constructions that would include a separate machine concept. The Plaintiff also seeks constructions that do not include the concept of Uniform Resource Locators (URLs) while the Defendants add the term UPL to some of the constructions they seek. Other confiict-ing claim construction positions are more specific to individual terms that are in dispute.

II. APPLICABLE LAW

"It is a 'bedrock principle' of patent law that 'the claims of a patent define the invention to which the patentee is entitled the right to exclude.' " *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed.Cir.2005) (en banc) (*quoting Innova/Pure Water Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed.Cir.2004)). In claim construction, courts examine the patent's intrinsic evidence to define the patented invention's scope. *See id.; C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 861 (Fed.Cir.2004); *Bell Atl. Network Servs., Inc. v. Covad Commc'ns Group, Inc.*, 262 F.3d 1258, 1267 (Fed.Cir.2001). This intrinsic evidence includes the claims themselves, the specification, and the prosecution history. *See Phillips*, 415 F.3d at 1314; *C.R. Bard, Inc.*, 388 F.3d at 861. Courts give claim terms their ordinary and accustomed meaning as understood by one of ordinary skill in the art at the time of the invention in the context of the entire patent. *Phillips*, 415 F.3d at 1312-13; *Alloc, Inc. v. Int'l Trade Comm'n*, 342 F.3d 1361, 1368 (Fed.Cir.2003).

The claims themselves provide substantial guidance in determining the meaning of particular claim terms. *Phillips*, 415 F.3d at 1314. First, a term's context in the asserted claim can be very instructive. *Id.* Other asserted or unasserted claims can also aid in determining the claim's meaning because claim terms are typically used consistently throughout the patent. *Id.* Differences among the claim terms can also assist in understanding a term's meaning. *Id.* For example, when a dependent claim adds a limitation to an independent claim, it is presumed that the independent claim does not include the limitation. *Id.* at 1314-15.

Claims "must be read in view of the specification, of which they are a part." *Id.* (quoting *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 978 (Fed.Cir.1995)). "[T]he specification 'is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.' " *Id.* (*quoting Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed.Cir.1996)); *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1325 (Fed.Cir.2002). This is true because a patentee may define his own terms, give a claim term a different meaning than the term would otherwise possess, or disclaim or disavow the claim scope. *Phillips*, 415 F.3d at 1316. In these situations, the inventor's lexicography governs. *Id.* Also, the specification may resolve ambiguous claim terms "where the ordinary and accustomed meaning of the words used in the claims lack sufficient clarity to permit the scope of the claim to be ascertained from the words alone." *Teleflex, Inc.*, 299 F.3d at 1325. But, "although the specification may aid the court in interpreting the meaning of disputed claim language, particular embodiments and examples appearing in the specification will not generally be read into the claims." *Comark Commc'ns, Inc. v. Harris Corp.*, 156 F.3d 1182, 1187 (Fed.Cir.1998); *see also Phillips*, 415

F.3d at 1323. The prosecution history is another tool to supply the proper context for claim construction because a patent applicant may also define a term in prosecuting the patent. *Home Diagnostics, Inc., v. Lifescan, Inc.,* 381 F.3d 1352, 1356 (Fed.Cir.2004) ("As in the case of the specification, a patent applicant may define a term in prosecuting a patent.").

*3 Although extrinsic evidence can be useful, it is "less significant than the intrinsic record in determining 'the legally operative meaning of claim language.' " *Phillips,* 415 F.3d at 1317 (*quoting* *C.R. Bard, Inc.,* 388 F.3d at 862). Technical dictionaries and treatises may help a court understand the underlying technology and the manner in which one skilled in the art might use claim terms, but technical dictionaries and treatises may provide definitions that are too broad or may not be indicative of how the term is used in the patent. *Id.* at 1318. Similarly, expert testimony may aid a court in understanding the underlying technology and determining the particular meaning of a term in the pertinent field, but an expert's conclusory, unsupported assertions as to a term's definition is entirely unhelpful to a court. *Id.* Generally, extrinsic evidence is "less reliable than the patent and its prosecution history in determining how to read claim terms." *Id.*

The patents in suit also contain means-plus-function limitations that require construction. Where a claim limitation is expressed in "means plus function" language and does not recite definite structure in support of its function, the limitation is subject to 35 U.S.C. § 112, ¶ 6. *Braun Med., Inc. v. Abbott Labs.,* 124 F.3d 1419, 1424 (Fed.Cir.1997). In relevant part, 35 U.S.C. § 112, ¶ 6 mandates that "such a claim limitation 'be construed to cover the corresponding structure ... described in the specification and equivalents thereof.' " *Id.* (citing 35 U.S.C. § 112, ¶ 6). Accordingly, when faced with means-plus-function limitations, courts "must turn to the written description of the patent to find the structure that corresponds to the means recited in the [limitations]." *Id.*

Construing a means-plus-function limitation involves multiple inquiries. "The first step in construing [a means-plus-function] limitation is a determination of the function of the means-plus-function limitation." *Medtronic, Inc. v. Advanced Cardiovascular Sys., Inc.,* 248 F.3d 1303, 1311 (Fed.Cir.2001). Once a court has determined the limitation's function, "the next step is to determine the corresponding structure disclosed in the specification and equivalents thereof." *Id.* A "structure disclosed in the specification is 'corresponding' structure only if the specification or prosecution history clearly links or associates that structure to the function recited in the claim." *Id.* Moreover, the focus of the "corresponding structure" inquiry is not merely whether a structure is capable of performing the recited function, but rather whether the corresponding structure is "clearly linked or associated with the [recited] function." *Id.*

III. DISCUSSION

A. Disputed Claim Terms

1. "*Web Page* "

"Web page" is utilized in asserted claims 1, 3, 6, 7, 9, and 11 of the '554 Patent and 1, 4, 7, and 8 of the '335 Patent. The Plaintiff asserts that "Web page" does not need construction. Alternatively, if construed, the Plaintiff asserts that the proper construction of the term is "content displayable through a Web browser." The Defendants assert that the terms should be construed as "an HTML document accessible through a URL."

*4 The Court first notes that the Plaintiff's original briefing expressed concern that the Defendants construction implies that dynamically generated Web pages are not included within the term "Web page." The Court finds this concern somewhat unfounded, and it is noted that the Defendants clearly referred to "Web page" in their briefing and oral argument as encompassing both static and dynamic Web pages.

The other assertions by the parties primarily revolve around two issues, the inclusion of terms "HTML" and "URL" in the construction. The Plaintiff argues that the term "content" is utilized in the specification at least twice to describe what is displayed on a Web page. Col. 1:47-51; Col. 7:23-26. Further, the Plaintiff points out that in the Defendants' own briefing Web pages are referred to as containing content. Defendants' Brief at 11-12. The Plaintiff further asserts that the Defendants are also attempting to read in limitations from the specification and that such a construction would exclude documents formatted in other formats such as SGML, XHTML, XML, and JPG.

The Defendants state that HTML is a software language and argue that as described within the specification HTML documents are what are sent back as Web pages. Col. 1:18-22; Figures 3 and 5. The Defendants also cite a Microsoft Press Computer Dictionary definition which states that "A Web page consists of an HTML file ..." Defendants' Brief at 9. The Defendants further assert that "content" in the specification refers to information included in a Web page and that such content itself does not form a Web page.

The Court notes that the specification does appear to consistently refer to the HTML language and does not mention other software languages. However, the Defendant does not identify persuasive support within the specification that the invention must be limited to only one type of software language. Moreover, Defendants have not persuaded the Court that in light of the specification one skilled in the art would assume a Web page as referred to in the patents could only be generated with the HTML language. Further, upon review of the whole specification and claims, with respect to Web pages the described concepts are not related to the intricacies of what particular programming languages are used to display a Web page but rather merely the higher level differentiation of static pre-existing Web pages verse dynamically generated Web pages. In these circumstances, the Court finds it improper to incorporate the limitation of HTML within the more general term Web page that is utilized in the claims themselves.

With regard to the URL concept, the Plaintiff asserts that the Defendants' definition adds additional complexity to the claim construction as the meaning and scope of "accessible through a URL" could itself require construction. Further, citing the extrinsic evidence that the Defendants themselves put before the Court, the Plaintiff argues that it is known that when a Web browser sends a request what is actually sent does not match what is commonly known as a full URL. In oral argument the Plaintiff asserted that the Defendants' extrinsic evidence shows a URL as "http_URL ="http:" "//" host [":" port] [abs_ path] thus requiring four components: a protocol (HTTP), a host, a port and an absolute path. Further the Plaintiffs argue that this same extrinsic evidence shows that a request most commonly is structured for example "GET / pub / WWW / TheProject.html HTTP / 1.0" The Plaintiff asserts that this also emphasizes the concern over the ambiguity of the meaning of "accessible through a URL."

*5 The Defendants turn to the specification, which includes the statement "[a] URL is a Web address that identifies the Web page and its location on the Web." Col. 1:30-33. The Defendants also note the language that states "[w]hen the appropriate Web site receives the URL, the Web page corresponding to the requested URL is located...." Col. 1:33-34. Further, the Defendants point to other examples in the specification, such as Figure 3, which refer to the Web browser sending the URL request. At oral argument, the Defendants also stated that what is sent by a Web browser does include the URL information.

The arguments of the parties highlight some of the concerns the Court has with the inclusion of the term URL. A definition of the meaning of URL within the usage of the patent would further be necessitated as URL is alternatively referred to as "what is examined by the Web browser," a URL request is received by the Web server, and a URL request can be sent to a page server. Col. 4:13- 14; Col. 8:28-32; Col. 8:38-39. The specification does not make clear what is the particular structure and content that is meant by the use of the term "URL" at each of these stages of the process. Again, as with the HTML term, this is not surprising as the specification and claims as a whole do not focus on the particular type of request that is made or the particular structure/content of a request as it processed through the system beyond the static and dynamic distinction discussed above. Further, to add the term "accessible" to the construction would necessitate further claim construction as to what "accessible" means. As described within the specification, a Web page is a mechanism through which static and dynamic content may be displayed. The particular addressing mechanism at each step of the processing of a dynamic Web page is not noted in the specification to be a requirement or of particular importance to the claimed invention. An inclusion of the term URL would improperly incorporate limitations from the specification for the term Web page which has a meaning that is adequately described within the full context of the specification. [FN2] **Thus, the Court construes "Web page" to mean "Web content displayable through a Web browser."**

> FN2. Additional discussion regarding the inclusion of URL limitations is also provided below with regard to the term "request ."

2. *"Request "*

"Request" is utilized in asserted claims 1, 9, 11 of the '554 Patent and 1, 15, and 29 of the '335 Patent. The Plaintiff asserts that the proper construction of "request" is "a message that asks for content." The Defendants assert that the term should be construed as "a message containing a URL that asks for a Web page specified by the URL."

The Court first notes that the term "request" generally appears in the claims in different circumstances. In some claims (for example claim 1 of the '554 Patent), request is first utilized with relation to "a dynamic Web page generation request" and multiple references are then made to "said request." Such claims also use the term "other requests" which imply a request different from "said request." Other claims (for example claim 15 of the '335 Patent) begin with the general use of "a request" but later refer to "dynamically generating a page in response to said request ." These claims also refer to "other requests." As noted in the specification, a request can refer to static Web pages (such as a static document) or dynamic Web pages (such as a Web page dynamically generated by an application). Col. 1:38-56, Col. 4:16-32 Thus, in its general use, request is not limited to dynamic or static requests. It is also noted that both types of claims consistently refer to either "Web page" generation or "page" generation. The specification uniformly refers to pages in the context of Web pages. Thus, it is not unreasonable that in light of the consistent specification one would interpret "requests" as relating to requests for Web pages.

*6 The primary point of distinction between the proposed constructions relates to inclusion of "URL" in the construction of request. The specification notes that "a Web browser allows a Web client to request a particular Web page from a Web site by specifying a Uniform Resource Locator (URL). A URL is a Web address that identifies the Web page and its location on the Web." Col. 1:29-32

The Defendants cites numerous places in the specification text in which "URL request" is utilized to refer to what is requested. The Plaintiff asserts that a request may be made at multiple points, such as shown in Figure 4 between the Web client and Web server, between the Web server and the Dispatcher and between the Dispatcher and the Page servers. The Plaintiff further argues that once a message is received by a Web server a full URL is not utilized subsequently, such as by Page servers.

The Defendants counter by noting numerous specification citations to the term "URL request" including "route the URL request to a Page server" and "the dispatcher sends the URL request to an appropriate Page server." Col. 6:9-10, Col. 8:38-39, *See* Defendants' Brief at 13-14. The Plaintiff responds that there is no teaching in the specification that a page server utilizes a URL. Moreover, the Plaintiffs assert that it would be illogical for a request provided to the page server to utilize a URL as such address would refer to the Web server.

The Court notes that within the claims "request" is used in context of multiple steps of the page generation process. For example, within claim 1 a request may be provided to a Web server while "said request" is also received by a Page server. The specification provides varied and not always consistent uses of the terms "request" and "URL request." As noted above, URL request is often utilized. However, the more general term "request" is also often utilized. Col. 2:1-12; Col. 2:18-35; Col. 4:33-53; Col. 5:8-59; Col. 6:20-32; Col. 7:5-6. In at least two of these instances, language stating "requests or 'hits' " is utilized. Col. 4:38-39; Col. 7:5-6. Further, the Defendants have not shown within the specification that a Page server utilizes a URL. Thus, even when "URL request" is provided in the specification it is not clear that such request is required to contain a URL or is merely a request generated from an initial URL provided at a Web client. To require "request" to include a URL would thus include limitations that the specification does not clearly support and clearly require.

The remaining dispute between the parties relates to the use of "content" verse "Web page." This dispute relates to the underlying meaning of the term Web page as discussed above and thus does not need to be re-addressed. As the Court has noted, the context of the patent is uniformly directed towards Web page requests. Under the guidance provided in *Phillips,* it is appropriate when viewing the specification and the language of the claims themselves to limit "request" to Web page applications. **Thus, the Court construes "request" to mean "a message that asks for a Web page" (with the term Web page having the construction provided herein)**

3. "*Page Server* "

*7 "Page server" is utilized in asserted claims 1, 4, 7, and 9-11 of the ' 554 Patent and 1, 2, 5, 8, 15-16, 19, 22 and 29 of the '335 Patent. The Plaintiff asserts that the proper construction of "page server" is "a processing system operable to receive a request and dynamically generate content in response to the request." The Defendants assert that the term should be construed as "page-generating software that generates a dynamic Web page on a machine separate from the Web server machine." In the claim construction Oral Argument, the Plaintiff agreed to the use of "page-generating software" in place of the term "a processing system" as previous proposed by the Plaintiff. The differences between the parties with regard to the use of "content" verse "Web page" are rooted in the basic dispute over the meaning of Web page as discussed above. Both parties include the concept of dynamic generation in their proposed constructions.

In the post oral hearing briefing, the parties each acknowledged that the primary dispute regarding the construction of "page server" is whether the Page server has to be on a machine separate from the Web server. As discussed below, the Court agrees with the Plaintiff with regard to this point of dispute.

Each party points to the specification to support their asserted position. The Plaintiff asserts that the specification includes statements that indicate that the Page server could operate on the same machine as the Web server. In particular, the Plaintiffs have pointed to passages which state:

> FIG. 1 illustrates a typical computer system 100 in which the present invention operates. Col. 2:66-67.
>
> The preferred embodiment of the present invention is implemented as a software module, which may be executed on a computer system such as computer system 100 in a conventional manner. Col. 3:55-58.

Figure 1 illustrates a computer system having a processor, bus, memory and mass storage. Further, it is stated that "the preferred embodiment of the present invention" may be implemented on a personal computer or alternatively a workstation. Col. 2:67-Col. 3:5. The Plaintiff asserts that this language is consistent with the specification as a whole by asserting that the specification describes a partitioned software architecture in which in some embodiments the software modules may all reside on the same machine and in other embodiments the software modules may reside on different computers.

The Plaintiff also points to a passage that describes an embodiment that does not have the advantage of "off-loading the processing of Web requests from the Web server machine" to a separate machine. Col. 5:26-36. However, the Court notes that this passage makes specific reference to the division between a Web server and a Dispatcher, and it is not clear in this passage alone that the Page server is also included in this use of a single machine.

The Defendants argue that the specification describes a distinction over the prior art that amounts to an explicit characterization of the invention that disclaims the prior art. *See SciMed Life Sys. V. Advanced Cardiovascular Sys.*, 242 F.2d 1337, 1343 (Fed.Cir.2001). In particular, the Defendants point to a passage of the specification that describes the multi-threading techniques of prior art Web servers. Col. 4:32-53. This passage concludes with "[t]he claimed invention addresses this need by utilizing a partitioned architecture to facilitate the creation and management of custom Web sites and servers." The Defendants assert that this clearly demonstrates that the purpose of the invention was to partition the various modules on separate machines. As to the passages cited by the Plaintiff, the Defendants assert those passages do not describe the entirety of the claimed invention. The Plaintiff asserts that the passage cited by the Defendants is directed toward the Web site management "need" recited in the passage, and this need is addressed by a partitioned architecture.

***8** The parties have thus each pointed to somewhat conflicting passages of the specification to support their positions. The passages cited by the Plaintiff establish that there is not a clear disavowal within the specification of the use of a partitioned software architecture on a single machine. The Defendants do correctly point to cases which stand for the proposition that when the specification makes clear that the invention does not include a particular feature than that feature is deemed to be outside of the reach of the claims of the patent. Defendants' Joint Sur-Reply, p. 8. However, in the specification before this Court, the specification does not make clear that the invention must only be operated on separate machines.

The Court construes "page server" to be "page-generating software that generates a dynamic Web page."

4. "*Web Server* "

"Web Server" is utilized in asserted claims 1 and 11 of the '554 Patent and 1-2 of the '335 Patent. The Plaintiff asserts that the proper construction of "Web server" is "a processing system capable of processing an HTTP request and producing a response to such a request." The Defendants assert that the terms should be construed as "a machine running a Web server executable capable of storing, locating, and returning Web pages in response to Web client requests." In the claim construction Oral Argument, the Plaintiff stated that it would agree to language including "software" in place of the Plaintiff's originally proposed "system" language similar to the Plaintiff's agreement with regard to "page server."

The focus point of the dispute between the parties is whether the term "Web server" requires a machine or whether the term may merely represent software or a combination of the two. Both the Plaintiff and Defendants cite conflicting extrinsic evidence to support their positions in the form of dictionaries, industry guides, and protocols. Some of the Plaintiff's extrinsic evidence includes citations to extrinsic evidence first brought before the Court by the Defendants. The conflicting extrinsic evidence presented by the parties fits the rationale presented in *Phillips* regarding the cautions that should be considered relating to such evidence.

Looking to the specification, the Plaintiff points to passages in which "Web server" is not used to describe a machine. In particular, the Plaintiff points to the statement in the specification that:

> The preferred embodiment of the present invention is implemented as a software module, which may be executed on a computer system such as computer system 100 in a conventional manner.

> Col. 3:55-58.

The Plaintiff also highlights the following passage:

> This embodiment is appropriate for Web servers such as Netsite TM from Netscape, that support such extensions. A number of public domain Web servers, such as NCSA TM from the National Center for Supercomputing Applications at the University of Illinois, Urbana-Champaign, however, do not provide support for this type of extension. Thus, in an alternate embodiment, Interceptor 400 is an independent module, connected via an 'intermediate program' to Web server 201. This intermediate program can be a simple CGI application program that connects Interceptor 400 to Web server 201. Alternate intermediate programs the perform the same functionality can also be implemented. Col. 4:63-Col. 5:7.

***9** The Defendants counter that the specification uses the terms "Web server," "Web server executable" and "Web server machine" and that the proper interpretation is that the machine is referred to as a Web server machine, the software is referred to as the "Web server executable" and the combination is referred to as a "Web server." To support this argument, the Defendants cite to various passages and figures in the specification. Figures 2-4; Col. 4:39-41; Col. 4:59-62; Col. 5:7-36.

While the Defendants may be correct that "Web server" may be utilized at times as indicating a combination of a machine and software, the specification clearly does not require the term "Web server" to include a machine. The passages of the specification noted above by the Plaintiff make clear that the Web server is contemplated to be at least in one embodiment, software. It is also noted that in general in other passages of the specification the term "Web server machine" is more often used when describing the machine component and "Web server" to describe merely the software component. Thus, for example, it is noted that the "Web servers process each of these requests on a single machine, namely the Web server machine," "Inteiceptor 400 resides on the Web server machine as an extension to Web server 201," and the Dispatcher "can, however, also reside on the same machine as the Web server." Col. 4:39-42; Col. 4:61-62; Col. 5:20-21. Passages such as these imply a utilization of the term "Web server" as the software module as opposed to the combination of both the machine and software. Thus, some usage of "Web server" implies just software and at other times implies a combination of software and hardware.

The Court construes "Web server" to be "software, or a machine having software, that receives Web page requests and returns Web pages in response to the requests.

5. "*HTTP-complaint device* "

"HTTP-complaint device" is utilized in asserted claims 15-16 and 19 of the ' 335 Patent. The Plaintiff asserts that "HTTP-complaint device" does not need construction. Alternatively, if construed, the Plaintiff asserts that the proper construction is "a device that understands HTTP and whose behavior is affected by an HTTP request." The Defendants assert that the terms should be construed as "a machine running an executable capable of storing, locating and returning Web pages in response to Web client requests."

The Defendants assert the same construction for the terms "Web server" and "HTTP-complaint device." The Defendants' construction is based upon their assertion that the term does not appear in the specification and that the specification only discloses a Web server for performing the function described in the language surrounding the use of "HTTP-compliant device." As such, the Defendants assert that "HTTP-compliant device" should be construed the same as "Web server" as that is the only corresponding device described and enabled in the specification. To hold otherwise, assert the Defendants, would result in a claim that is overbroad and invalid for not being described and enabled.

***10** Regarding the maxim that claims should be construed to be valid, in *Phillips* the Federal Circuit guidance states that this maxim is limited "to cases in which 'the court concludes, after applying all the available tools of claim construction, that the claim is still ambiguous.' " Phillips, 415 F.3d at 1327. The Court does not find that in light of the specification the term in question is ambiguous to one skilled in the art. Thus, the Defendants' validity concerns should be more properly addressed with regard to validity motions.

The Defendants also assert that the more general construction proposed by the Plaintiff would be so broad as to even encompass a Web client. However, the claim language itself makes clear that this concern is not valid as there is substantial functional language regarding what happens at the HTTP-compliant device including the transferring of a request from the HTTP-compliant device to a page server, intercepting the request at the HTTP-compliant device, and concurrently processing other requests at the HTTP-compliant device.

The Defendants do however raise valid concerns over the Plaintiff's definition raising additional interpretation questions with regard to the meaning of "understands HTTP" and "behavior affected by an HTTP request." The Court agrees with the Defendants in this regard. The specification defines HTTP as "a communications protocol known as Hyper-Text Transport Protocol (HTTP) ." Col. 1:25-26. Mindful that not all terms in a claim need construction, **the Court adopts a construction of "HTTP-compliant device" to mean "a device that is compliant with the communication protocol known as HyperText Transport Protocol (HTTP)."**

6. "*Said processing being performed by said page server while said Web server concurrently processes said other requests* "

"Said processing being performed by said page server while said Web server concurrently processes said other requests" is utilized in asserted claims 1 and 11 of the '554 Patent and 1, 15 and 29 of the '335 Patent. The Plaintiff asserts that the "said processing ..." phrase does not need construction. Alternatively, if construed, the Plaintiff asserts that the proper construction is "said processing being performed by said page server while said Web server processes said other requests at the same time." The Defendants assert that the phrase should be construed as "said processing being performed by said page server while said Web server executable processes other requests literally at the same time on a different machine."

Three main points of distinction exist between the parties: the inclusion of "executable" with regard to the use of Web server, the inclusion of "literally" with regard to the "same time" language, and the inclusion of the concept of the Page server and Web servers being on different machines. With regard to the term "Web server" as utilized within the "said processing ..." phrase, the Court finds that the construction the Court provided above for "Web server" is applicable, and thus the inclusion of the term executable does not need to be re-addressed with relation to the "said processing ..." phrase. Similarly, with regard to the concept of different machines, the Court has addressed that concept above and does not need to re-address that concept here.

***11** With regard to "concurrently," both parties agree that this term includes the concept of something occurring "at the same time;" however, there is still a dispute as to whether it must be "literally at the same time." The Defendants argue that "concurrently" should be analyzed in the context of the discussion in the patents of the prior art time-interleaved multi-threading techniques. Accordingly, the Defendants argue that if concurrently is not read to be "literally at the same time" the claims would read on time-interleaved multi-threading techniques. Once again, such arguments are more suited for invalidity assertions. Moreover, it is noted that in the passage in which the patents utilize the term "concurrently" to describe the Web server and the Page server operations, the patents also describe this concept as "to simultaneously process." Col. 6:21-27. With regard to the prior art time-interleaved multi-threading discussion (which the Defendants assert is not literally at the same time) the patent also uses the term "simultaneously." Col. 4:48-51. The patents do not distinguish one use of the term simultaneously from the other by the inclusion of the literal concept. The Court does not find support in the intrinsic evidence to support a requirement that "at the same time" must be "literally at the same time." **Thus, the**

court construes "said processing being performed by said page server while said Web server concurrently processes said other requests" to mean "said processing being performed by said page server while said Web server processes said other requests at the same time."

7. "*Intercepting* "

"Intercepting" is utilized in asserted claims 1, 9, 10 and 11 of the '554 Patent and 1 and 15 of the '335 Patent. The Plaintiff asserts that the proper construction of "intercepting" is "stopping, deflecting, or interrupting the processing of a request." The Defendants assert that the term should be construed as "diverting a request received at the Web server machine instead of the Web server executable processing it." Both parties argue that the other party's interpretations carry implicit meanings beyond the mere constructions asserted above. The Plaintiff asserts that the Defendants' construction implicitly requires a request to go around or bypass a Web server without any processing by the Web server. The Defendants assert that the Plaintiff's construction allows for the Web server executable to begin processing a request, even if only to recognize that is should not complete the processing. The parties also disagree as to whether the terms "stopping, deflecting, or interrupting" verses "diverting" are more appropriate. A portion of the disagreement between the parties is based upon the fundamental dispute as to whether a Web server is a machine, software, or combination thereof. As noted above, the Court construes a Web server as software, or a machine having software.

***12** The Defendants assert that the prosecution histories of the '554 Patent and of the '335 Patent add clarity to the meaning of the term "intercepting." The Defendants are correct that prosecution history may be used to limit the claims so as to exclude interpretations that may have been disclaimed or disavowed. However as discussed below, the cited portions of the prosecution histories of the '554 Patent and the '335 Patent do not amount to a clear disclaimer as suggested by the Defendants. *See Middleton, Inc. v. Minnesota Mining & Mfg. Co.,* 311 F.3d 1384, 1388 (Fed.Cir.2002).

With regard to the '554 Patent file history, the Defendants point to language added via an Examiner's Amendment and to the Bookman reference for support of a disclaimer of claim scope. The amendment in question was made in a Notice of Allowability that was issued along with an Interview Summary of a December 17, 1998 examiner interview and a form PTO-892 Notice of References Cited. The PTO-892 Notice of References Cited listed the Bookman reference and an additional reference. At that time, other art was also included in the prosecution history including, for example, references that were the basis of previous rejections. The Interview Summary merely states that the prior art discussed was the "prior art of record." The description and other comments of the interview do not provide any other details as to why the Examiner Amendment was made. The Amendment in question added, to claim 1 for example, the language "wherein said routing step further includes the steps of intercepting said request at said Web server, routing said request from said Web server to a dispatcher, and dispatching said request to said page server." As the Plaintiff points out, the record does not show that Bookman was even particularly discussed in the interview let alone the art requiring the amendment to be made. On this basis alone, the Court may reject the Defendants' assertion that because of Bookman the claim language must be interpreted in the manner that the Defendants allege. Further, even if Bookman had been the art requiring such amendment, the record still does not provide any insight into the meaning of "intercepting" as the record is silent as to any further meaning of "intercepting" or the other additional terms recited in the amendment (routing from a Web server to a dispatcher and dispatching said request to a page server).

With regard to the '335 Patent prosecution history, the Defendants assert that statements made by the Applicants regarding the Leaf reference support the Defendants' proposition that partial processing does not equate to intercepting. In particular, the Defendants cite to a quote on pages 9-10 of a Response To Office Action dated May 23, 2001. The

Defendants focus on a statement that Leaf did not suggest intercepting because "merely routing a request from a web server to the transaction gateway does not involve interception." The Defendants state that Leaf has Web server executable that partially processes a request before routing it on to the dispatcher. From this, the Defendants argue that the Applicants disclaimed partially processirg a request and then routing it to the dispatcher. It is noted that the Applicants' Response in question, however, does not characterize Leaf in the manner suggested by the Defendants or make any references to partially processing. Further, the distinction between a Web server and Web server executable made by the Defendants is not clear in Leaf. In addition, it is noted that the full context of the Applicants' remarks regarding Leaf includes the statement that "Leaf does not teach or suggest 'intercepting said request.' Instead, Leaf teaches that the web server routes the request directly to the transaction gateway client." '335 Patent File History, Response to Office Action Dated May 23, 2001, p. 9-10. This statement merely suggests that directly routing a request from a web server to a transaction gateway is not intercepting and does not provide clear guidance as to the question of partial processing or the difference between a Web server and Web server executable. Thus, the prosecution history cited by the Defendants does not provide the clear guidance asserted by the Defendants.

***13** It is noted that the Defendants' proposed construction adds Web server machine to the term of "intercepting." This language is somewhat redundant with the language surrounding the term "intercepting" in most claims and does not conform to claim 15 of the '335 Patent which uses the term HTTP-complaint device. The context of the term as used in the claims themselves provides some guidance as to the proper construction. The claims themselves note that the intercepting of the request is at the Web server or the HTTP-compliant device. [FN3] For example, Claim 1 uses language such as "routing said request from said Web server to a page server," and "wherein said routing step further includes the steps of intercepting said request at said Web server, routing said request from said Web server to a dispatcher, and dispatching said request to said page server." This conforms with the specification, which states that the request is initially routed from the Web Client 200 to the Web Server 201. Col. 4:54-60. Also, Figure 4 appears to show the request going to the Web server executable 201(E). A construction that has the request bypassing the Web server is therefore not appropriate.

FN3. Claim 15 of the '335 Patent uses the term HTTP-compliant device while the other claims recite a Web server.

As to whether the beginning phrase of the construction should include "stopping, deflecting, or interrupting" as proposed by the Plaintiff, the Plaintiff provides little support for such language other than a general purpose dictionary. Although the Plaintiff asserts that the Defendants' construction requires bypassing the Web server, the Court does not interpret the phrase "diverting" to require such bypassing. However, the term "diverting" seems to carry an additional connotation that the request is sent somewhere else. When looking at the claims, however, the concept of the request being sent elsewhere is included in the "routing said request" (claims 1, 9, and 11 of the '554 Patent) or "transferring said request" (claim 15 of the '335 Patent) limitation that immediately follows the intercepting phrase. Thus, the Court does not feel that either construction adds clarity to the meaning of the concept of "intercepting" as used in the claims. Moreover, the parties have not pointed to anything in the intrinsic record that suggests which is more accurate: "diverting" or "stopping, deflecting or interrupting." These terms are not used in the specification and each in turn may need their own construction. The Court is not convinced that the term "intercepting" needs construction itself or that the constructions proposed by the parties add any needed clarity.

More helpful would be to construe the entire intercepting phrase: "intercepting said request at said Web server" (claims 1, 9, and 11 of the ' 554 Patent) and "intercepting said request at said HTTP-compliant device" (claim 15 of the '554 Patent).

The specification describes the Interceptor as intercepting "the handling of a request." Col. 8:31-32. To conform with the description provided within the specification, the phrase "intercepting said request at said Web server" (claims 1, 9, and 11 of the '554 Patent) means at least "intercepting the handling of a request at a Web server" and the phrase "intercepting said request at said HTTP-compliant device" means at least "intercepting the handling of a request at a said HTTP-compliant device."

***14** What is left for the Court to determine is whether the phrase "instead of the Web Server executable processing it" should be added to the end of the definition as proposed by the Defendants. The Defendants' primary support for their position is the language of the specification that states "instead of Web server executable 201(E) processing the URL request, however, interceptor 400 intercepts the request and routes it to Dispatcher 402." Col. 4:58-60. The Defendants argue that this language provides no room for partial processing of a request by Web server executable. The Court, however, finds such language ambiguous. The Defendants would like to interpret this cited quote ("instead of the Web server executable 201(E) processing the URL request") to mean "instead of the Web server executable 201(E) processing **any of** the URL request" wherein the Plaintiff would like to interpret this citation to mean "instead of the Web server executable 201(E) **completely** processing the URL request" The specification, however, provides little guidance.

As discussed above, however, the specification does establish that a Web server may be software. Further, the specification establishes that the Web server does perform at least some action with relation to a request, namely receiving a request. Col. 4:55-58; Col. 8:28-31; Figure 4; Figure 5. Further, it is noted that in at least one embodiment the interceptor 400 is "an extension of the Web server 201" and the interceptor 400 also performs actions on a request. Col. 4:59-62. As Defendants have asserted that their proposed claim langue would exclude the Web server from any processing of the request, their proposed claim language would impermissibly exclude the embodiments disclosed within the specification. Thus, the Court declines to add the additional limitation sought by the Defendants.

For these reasons, **the Court finds that "intercepting said request at said Web server" means "intercepting the handling of a request at a Web server" and the phrase "intercepting said request at said HTTP-compliant device" means at least "intercepting the handling of a request at a said HTTP-compliant device."**

8. *"Transferring "*

"Transferring" is utilized in asserted claims 15-16 and 29 of the '335 Patent. The Plaintiff asserts that the "transferring" does not need construction. Alternatively, if construed, the Plaintiff asserts that the proper construction is "sending." The Defendants assert that the term should be construed as "sending toward a destination and relinquishing control of."

The parties agree to the use of the term of "sending" but disagree as to whether any additional language is necessary. With regard to the Defendants' use of "toward a destination," it is noted that the surrounding claim language of each claim already includes this concept. For example, claim 15 states that the transferring is of "a request from an HTTP-complaint device to a page server." Similarly, claim 29 states that the transferring is of "a request from an HTTP-complaint device to a dispatcher." The inclusion of "toward a destination" within the term "transferring" itself is therefore unnecessary based upon the context of the claims themselves.

***15** With regard to the "relinquishing control of" language sought by the Defendants, the Defendants point to a Microsoft Press Computer Dictionary definition that includes in transferring the concept of passing program control. Further, the Defendants assert that the purpose of the patent is to reduce processing burden. The Plaintiff argues that two other dictionary definitions noted by the Defendants do not include the relinquishing concept.

Once again, it is more instructive to look to the claims themselves. More particularly, in claim 15 of the '335 Patent immediately after the "transferring ... to a page server" the claim continues with "said page server receiving said request and releasing said HTTP-compliant device to process other requests...." Likewise, claim 29 of the '335 Patent includes later in the claim "said page server receiving said request and releasing said HTTP-compliant device to process other requests ..." To include relinquishing within the definition of transferring would possibly conflict with the latter claim language or alternatively be redundant. Further, to the extent that the Defendants assert that the purpose of the patent is to reduce processing burden, the recited releasing language is more related to that concept then the transferring.

As both parties have proposed definitions using the term "sending," the Court includes that term in its construction. **The Court construes "transferring" to mean "sending."**

9. "*Dispatching* "

The parties originally proposed different constructions for the term "dispatching." As the parties have subsequently submitted an agreed construction, a Court construction is no longer required.

9. "*Releasing* "

"Releasing" is utilized in asserted claims 1, 9 and 11 of the '554 Patent and 1, 15, and 29 of the '335 Patent. The Plaintiff asserts that the "releasing" does not need construction. Alternatively, if construed, the Plaintiff asserts that the proper construction is "freeing." The Defendants assert that the term should be construed as "communicating to said Web server that it may now process other requests." [FN4] A central point in the dispute between the parties is the Plaintiff's arguments that releasing can implicitly occur as a result of routing without communication to the Web server. The Defendants assert that there must be communication to the Web server.

The Plaintiff asserts that the term "freeing" is supported by the specification and notes that the specification states that the result of routing is that the Web server is free to continue servicing client requests. In particular, the Plaintiff points out that the specification states "Web server executable 201(E) is thus free to continue servicing client requests on Web serer 201 while the request is processed 'off-line.' " Col. 5:16-18. Thus, the Plaintiff asserts that the specification implies that releasing is an automatic consequence of routing the request to another processing element and that nothing in the specification requires communication to the Web server to effectuate the release.

***16** The Defendants assert that the '335 Patent prosecution history shows that merely routing a request from a Web server to a Page server and thereby implicitly releasing the Web server was disclaimed by the Applicants during the prosecution history as being different from releasing. The Plaintiff counters by asserting that the full context of the prosecution history quote in question does not stand for the proposition asserted by the Defendants. The portion of the prosecution history in question includes:

> At no time does Rogers teach or suggest 'concurrently' processing other requests or 'releasing said Web server to process other requests' because merely retrieving data from multiple sources does not teach or suggest these elements. Response to Office Action, Nov. 27, 2001 at 9.

The Court is persuaded that by reviewing the full context of the portion of the prosecution history in question a clear disavowal was not made by the Applicants. Thus, the prosecution history does not mandate that releasing cannot implicitly occur due to routing from a web server to a page server.

However, the Defendants raise a more relevant argument with regard to the claim language itself. "Releasing" is used in each claim as part of the phrase

FN4. For claim 9 of the '554 Patent, the Defendants substitute "second computer system" in place of "Web server" and similarly in claims 15 and 29 of '335 Patent the Defendants substitute "HTTP-compliant device" in place of "Web server."

"said page server receiving said request and releasing said Web server to process other requests." [FN5] It is therefore the Page server that does the releasing. The larger context of the use of releasing in the claims themselves indicates that the Page server has a role in the releasing as in the claims themselves it is the Page server that releases the Web server. This language also conforms to the Summary of Invention and Abstract. Col. 2:25-26; Abstract, line 8.

> FN5. "Second computer system" is used rather than "Web server" in claim 9 of the '554 Patent. "HTTP-compliant device" is used rather than "Web server" in claims 15 and 29 of '335 Patent.

The Court is thus persuaded that the Defendants are partly correct that as required by the claim language itself the Page server takes some action to releasing the Web server. However, the Defendants do not provide adequate support in the specification or elsewhere to mandate that such action is limited to communication from the Page server to the Web server. As there could be other actions that the Page server may take to affirmatively release the Web server, it would be inappropriate to limit the claim to one type of action, particularly when support is lacking in the record for that particular type of action. The specification does not necessarily limit the claims to a particular technique by the Page server as to how the claimed release by the Page server is accomplished. For example, the limitations proposed by Defendants could be argued to not include Page server actions that are communications from the Page server to other intermediate elements (such as the Dispatcher) or Page server actions that involve affirmatively not sending some expected regular communication. **The Court therefore interprets "said page server receiving said request and releasing said Web server to process other requests" to mean "said page server receiving said request and said page server performing an act (separate from merely receiving the request) to free the Web server to process other requests." Claims 9 of the '554 Patent and claims 15 and 29 of the ' 335 Patent are likewise construed with the substitution of "second computer system" and "HTTP-compliant device" respectively for "Web server."**

B. Means Plus Function Elements

*17 The parties propose different constructions of three means plus function elements. In each case, the parties agree to the function but disagree as to what is the corresponding structure that is disclosed in the specification pursuant to 35 U.S.C. § 112, ¶ 6. The following means plus function elements contained in claim 9 of the '554 Patent are disputed:

(a) "means for generating said request,"
(b) "means for receiving said request from said first computer" and
(c) "page server processing means processing said requests and dynamically generating a web page in response to said request."

Throughout the pre-hearing and post-hearing briefing, the parties have substantially changed their positions as to the appropriate legal standards and the appropriate structure that should be applied to each element.

As to the function, the parties have agreed to functions that match the claimed language recited above: "generating said request," "receiving said request from said first computer," and "processing said requests and dynamically generating a web page in response to said request" respectively.

The Plaintiffs propose that the corresponding structure for each means plus function element is "any software, processor or equivalent therefore that performs the function of ----" (where the blank is the agreed function for that means plus function element). The Defendants in contrast point to particular structure disclosed within the specification. For the "means for generating," the Defendants propose "a processor, such as Box 102 of Figure 1, on a Web client 200 of Figure 4, running a Web browser." For the "means for receiving," the Defendants propose "Web server 201 of Figure 4 running a web server executable." For the "page server processing means," the Defendants propose "a processor, such as Box 102 of Figure 1, on a page server machine, such as Box 404(1)-(n) of Figure 4, running undisclosed page-generating software ."

First, the Court notes that means plus function elements shall be construed to cover the corresponding structure described in the specification a.1d equivalents thereof. 35 U.S.C. § 112, ¶ 6. To avoid confusion, the Plaintiff indicated that it did not object to removing the word "equivalents" internal to its proposed construction. Plaintiff's Reply Brief at 58. The Court agrees that it would be clearer for "equivalents" to modify the entire construed structure.

As proposed by the Plaintiff, the corresponding structure could be any software or could be any processor that performs the agreed function. As proposed by the Plaintiff, software would not even be required as the Plaintiff uses the "or" conjunction. Plaintiff's construction of any software or processor is essentially unbounded and contradicts the guidance that has been established for interpreting means plus function elements. *Medical Instrumentation and Diagnostics Corp. v. Elekta AB,* 344 F.3d 1205, 1211 (Fed.Cir.2003)("We cannot allow a patentee to claim in function terms essentially unbounded by any reference to what one skilled in the art would understand from the public record.") Having chosen to utilize the means plus function claiming structure, the Plaintiff cannot now avoid its implications. Moreover, for computer implemented implementations the corresponding structure for means elements is limited to the specific structure and specific algorithms disclosed in the specification as opposed to the generic "software" or "processor" as asserted by the Plaintiff. *See Harris Corp. v. Ericsson Inc.,* 417 F.3d 1241, 1253-54 (Fed.Cir.2005); *WMS Gaming Inc. v. International Game Technology,* 184 F.3d 1339, 1348-49 (Fed.Cir.1999). Thus, the court must look to the specific structures and algorithms disclosed with the specification.

***18** As noted above, the specification states:

> FIG. 1 illustrates a typical computer system 100 in which the present invention operates. Col. 2:66-67.
>
> The preferred embodiment of the present invention is implemented as a software module, which may be executed on a computer system such as computer system 100 in a conventional manner. Col. 3:55-58.

As described within the specification, the various software components operate on computer systems such as computer system 100 of Figure 1 that includes a processor 102. [FN6] Col. 2:66-67; Col. 3:8-11; Col. 3:55-63.

> FN6. It is noted that as to the component of the computer system, both parties agree that reference to the processor is appropriate.

1. *"means for generating said requests,"*

For example, the "means for generating a request" is described as a Web client machine running a Web browser. Col. 1:24-26; Col. 2:4-7; Col. 4:12-15; Col. 4:55-57; Col. 6:27-31; *See* Col. 8:25-29. This particular structure is noted in the specification for accomplishing the claimed function. One skilled in the art would understand from the specification that a Web client computer having a processor which operates a Web browser is the corresponding structure that accomplishes the function of generating a request.

The Court construes the corresponding structure of the "means for generating" to be "a processor of a computer that is, or has, a Web client running a Web browser" or equivalents thereof.

2. *"means for receiving said request from said first computer "*

With regard to the "means for receiving said request from said first computer" element, the specification shows the Web server 201 receiving requests from the Web client 200. Figure 4; Figure 5; Col. 4:54-59; *See* Figure 2; Col. 3:64-Col. 4:10. The Web server 201 is also repeatedly described as including Web server executable. *Id.* As described above, the specification also establishes that that even if a Web server is software, the software module operates on a computer as described within the specification.

The Court construes the corresponding structure of the "means for receiving" to be "a processor of a computer that is, or has, a Web server running Web server executable" or equivalents thereof.

3. *"page server processing means processing said requests and dynamically generating a web page in response to said request."*

With regard to the function of the "page server processing means processing said requests and dynamically generating a web page in response to said request" element, the specification shows that such function is accomplished by Page server 404(1)-(n). Figure 4; Figure 5; Col. 5:37-Col. 6:31; Col. 8:39-43. As noted above, the parties agree that a Page server is page generation software. Further, as also noted above with the other means plus function elements, the specification establishes that software is operated on a computer system as described with the specification. Moreover, with regard to Page servers, it is noted that in one embodiment Page servers reside on a separate machine to accomplish the claimed function. Col. 5 line 49-50. Thus, the Court finds that the Page server processing means is Page server software (as construed above) operating on a computer processor.

***19 The Court construes the corresponding structure of the "page server processing means" to be "a processor of a computer that runs Page server software (wherein Page server software is page-generating software that generates a dynamic Web page)."**

IT IS SO RECOMMENDED.

Within ten (10) days after receipt of the magistrate judge's report, any party may serve and file written objections to the findings and recommendations of the magistrate judge. 28 U.S.C.A. 636(b)(1)(C).

Failure to file written objections to the proposed findings and recommendations contained in this report within ten days after service shall bar an aggrieved party from *de novo* review by the district court of the proposed findings and recommendations and from appellate review of factual findings accepted or adopted by the district court except on grounds of plain error or manifest injustice. *Thomas v. Arn,* 474 U.S. 140, 148 (1985); *Rodriguez v. Bowen,* 857 F.2d 275, 276-77 (5th Cir.1988).

Not Reported in F.Supp.2d, 2006 WL 3099603 (E.D.Tex.)

END OF DOCUMENT

H

Only the Westlaw citation is currently available.

United States District Court,
D. Oregon.
COLLEGENET, INC., a Delaware corporation,
Plaintiff,
v.
XAP CORPORATION, a Delaware corporation,
Defendant.
No. CV-03-1229-HU.

Oct. 29, 2004.

Kristin L. Cleveland, Scott E. Davis, Jared S. Goff, Michael N. Zachary, Klarquist Sparkman, LLP, Portland, Oregon, for Plaintiff.

Alexander C. Johnson, Stephen S. Ford, Marger, Johnson & McCollum, P.C., Portland, Oregon, Daniel Johnson, Jr., Henry C. Su, Fenwick & West LLP, Mountain View, California, for Defendant.

FINDINGS & RECOMMENDATION
HUBEL, Magistrate J.

*1 Plaintiff CollegeNET, Inc. brings this patent infringement action against defendant XAP Corporation. Plaintiff is the owner of two patents: United States Patent Number 6,345,278 B1 ("the '278 patent), and United States Patent Number 6,460,042 ("the '042 patent). In its Second Amended Complaint, plaintiff brings two claims of patent infringement, one for each of the two patents. Plaintiff also brings a claim for declaratory judgment related to a September 10, 2003 press release published by plaintiff, as well as two claims for unfair competition.

Defendant raises affirmative defenses of noninfringement, invalidity, and unenforceability. Defendant additionally counterclaims for declaratory judgments of noninfringement, invalidity, and unenforceability. Finally, defendant also brings claims for unfair competition.

Presently, the parties seek construction of various terms of the two patents. This Findings & Recommendation contains my recommended claim constructions.

This Court has had the opportunity to previously oversee the adjudication of these two patents in two cases, unrelated to the present case, brought by plaintiff against ApplyYourself, Inc. In case number CV-02-484-HU, plaintiff brought infringement claims against ApplyYourself related to the '278 patent. In case number CV-02-1359-HU, plaintiff brought infringement claims against ApplyYourself related to the '042 patent. The cases were consolidated and were tried to a jury in August and September 2003. As part of that litigation, I construed some claims in both patents. *CollegeNET v. ApplyYourself*, No. CV-02-484-HU, Opinion (D.Or. Dec. 19, 2002) (construing claims in the '278 patent); *CollegeNET v. ApplyYourself*, Nos. CV-02-484- HU, CV-02-1359-HU, Opinion (D.Or. July 7, 2003) (construing claims in both patents). I refer to my previous constructions as discussed below.

BACKGROUND AND OVERVIEW OF THE INVENTIONS

I. The '278 Patent

As described in the patent itself, students applying to colleges and universities typically complete a separate paper application for each institution to which they seek admission. Exh. A to Sec. Am. Compl. (Col. 1, lines 19-21). [FN1] The applicant then mails each application to the corresponding institution along with a fee. 1:21-23.

Many institutions prefer Internet applications. 1:24-25. One problem with such applications, however, is that the student is required to re-enter the same information for each subsequent application to a different institution or to the same institution perhaps for a different academic term. Addi-

[FN1] References to the '278 patent will be to this exhibit and will be denoted simply by the column and line number referred to, such as 1:19-21.

tionally, the institution cannot change the application form without revising the source code that creates the application form, making changes expensive and inconvenient. 1:26-34.

One way to reduce the redundancy for the applicant would be to allow students to complete a single, generic application form provided by a third party who would then transmit the application to any designated institution. 1:35-38. The drawback to such a system is that the institution cannot customize its application form. 1:38-48.

***2** As described by plaintiff, a typical applicant would use the patented invention by first viewing a college's website and proceeding to the admissions web page. The student would typically be using a personal computer that was running a web browser to access the website. Somewhere in the on-line admissions materials, there is a prompt for applying on line. Clicking on this would lead the student through a series of instruction pages and ultimately to a "log on" page where the student establishes a user ID and a password.

Next, the student would receive a prompt which would say something like "Application" or "Apply Now." When the student clicks on this prompt, the browser on the student's computer sends a request over the Internet to the third party servicer's web server. The forms engine processes the request. The request itself would identify the specific college application form being requested and the student who is requesting it.

The forms engine then creates a copy of the requested application form for the student. The forms engine can create a copy of the college's application form in a variety of ways. The forms engine queries the database to determine whether the particular student has information stored that corresponds to any of the fields in the newly requested application form. If it is that student's first application with the third party servicer's system, there will be no stored information. If the student has previously filled out other application forms, the forms engine will identify the like fields and obtain data from the database. The forms engine will merge the retrieved data into the corresponding form data fields on the application form. The forms engine will then provide the application form to the student, sending it from the web server over the Internet.

The student then enters personal information into the form data fields. When the student clicks "save" or "save and go to next page," the browser will send the information entered by the student and post it to the third party servicer's web server. The forms engine then stores the information into the database.

Once the student has completed the form, the student can hit the "transmit" or "send application to school" prompt. It is then possible for the forms engine to perform a "data validation" check on the application. For example, Lewis & Clark College may specify that the "high school attended" and "SAT scores" are required fields and that it will not accept application forms in which those fields are left blank.

In addition, error-checking criteria may be specified, such as the "SAT score" must be between 200 and 800. The forms engine compares the data entered by the student for these fields and determines whether they are filled in and whether any specified criteria are met. If the criteria are met, the data is "valid" and will be further processed. If not, the system will send back to the student, over the Internet, an error-correction form or message that the student must change entries for the fields that did not meet the prescribed criteria.

***3** Once the application is complete, the student can also select a payment method to "e-pay" the school's application fee.

Next, the same student may want to fill out a second application to a different college or university. The student logs on to that school's website and follows the application prompts. The forms engine creates an application form for the student. Assuming both colleges use plaintiff's services, and because the student has previously filled out a form, information regarding that student is already stored in the database. The forms engine will retrieve information required for the second application that the student has already entered on the first application. The forms engine then automatically inserts the previously stored information required by the second application, into the form data fields of the second application form and sends it back to the student over the Internet.... The student will fill in the remaining blanks of the form, then "save" it. His or her data will then be sent over the Internet and posted to the web server. The forms engine will then store the data in the database.

II. The '042 Patent

The '042 patent is a continuation patent of the '278 patent. Exh. B to Sec. Am. Compl. (Col. 1, lines 4-5). [FN2]

> FN2. References to the '042 patent will be to this exhibit and will be denoted simply by the column and line number referred to, such as 1:4-5.

The abstract of the patent explains the patent as follows:

> A forms engine allows data sharing between customizable on-line forms, such as college admissions applications. Before applying, an applicant opens an account with a third party application servicer. After the applicant completes an application for one institution, the data is saved in a data base and automatically populates fields in subsequent application forms. The form for each institution is created from a form description file. Each form is branded for its institution and forms for different institutions differ in appearance and content so that the presence of the third party servicer is transparent to the applicant.
>
> The system is extensible without programming, allowing new applicant attributes to be readily incorporated into the system and allowing the content and appearances of the application to be readily changed by changing the description file. The use of aliases for applicant attributes permits data to be readily shared between forms even though labeled and arranged differently on different forms. Information stored about each attribute allows the specification of data validation rules and data sharing and grouping rules, as well as dependency rules that permit application page content to depend on applicant's responses on a previous page.

Exh. B to Sec. Am. Compl. at p. 1.

The attributes of the '042 patent are fairly consistent across the independent claims and include the following general features: (1) presenting a customized form to an applicant; (2) allowing the applicant to enter user and payment information; (3) receiving the user and payment information; (4) processing the user and payment information; and (5) sending the user information back to the institution in a format specified by the institution.

***4** The dependent claims further define the form in various ways: (1) as having multiple pages, as seen in claims 6, 21, 33, and 43; (2) providing for data validation at the client computer or after each page of the multiple page application is posted, as seen in claims 7, 8, 10, 11, 14, 22, 23, 25, 27, 30, and 33; (3) providing for further data validation at the server level or when the application is completed, as seen in claims 8, 10, 11, 12, 14, 23, 25, 27, 28, 30, 34, and 35; and (4) providing for automatic data population between multiple application forms, as seen in claims 4, 19, 36, 37, and 41.

CLAIM CONSTRUCTION STANDARDS

The first step in any validity or infringement analysis is to construe the claims. *See, e.g.,* *Smiths Indus. Med. Sys., Inc. v. Vital Signs, Inc.,* 183 F.3d 1347, 1353 (Fed.Cir.1999) ("the first step in any validity analysis is to construe the claims of the invention to determine the subject matter for which patent protection is sought"); *Markman v. Westview Instruments, Inc.,* 52 F.3d 967, 976 (Fed.Cir.1995) (en banc) (first step in two-step patent infringement analysis is to determine "the meaning and scope of the patent claims asserted to be infringed[, ...] commonly known as claim construction or interpretation[.]"), *aff'd,* 517 U.S. 370 (1996). The meaning of a term in a patent claim is a matter of law to be resolved by the court. *Markman,* 517 U.S. at 389-91.

Claims should be interpreted, when reasonably possible, to preserve their validity. Modine Mfg. Co. v. United States Int'l Trade Comm'n, 75 F.3d 1545, 1556 (Fed.Cir.1996). In construing a claim, the court should first look to the intrinsic evidence, that is, the claims themselves, the written description portion of the specification, and the prosecution history. Bell & Howell Document Mgmt. Prods. Co. v. Altek Sys., 132 F.3d 701, 705 (Fed.Cir.1997).

Generally, claim construction begins with the words of the claim. K-2 Corp. v. Salomon S.A., 191 F.3d 1356, 1363 (Fed.Cir.1999).

> It is standard practice that in determining the proper construction of an asserted claim, the court looks first to the intrinsic evidence--the patent specification, including of course the written description, and, if in evidence, the prosecution history. Absent an express definition in the specification of a particular claim term, the words are given their ordinary and accustomed meaning; if a term of art, it is given the ordinary and accustomed meaning as understood by those of ordinary skill in the art.

Zelinski v. Brunswick Corp., 185 F.3d 1311, 1315 (Fed.Cir.1999); *see also* Georgia-Pacific Corp. v. United States Gypsum Co., 195 F.3d 1322, 1332 (Fed.Cir.) ("The specification of the patent in suit is the best guide to the meaning of a disputed term."), *amended*, 204 F.3d 1359 (Fed.Cir.1999).

Terms in a claim are given their ordinary meaning to one skilled in the art unless it appears from the patent and prosecution history that the inventor used them differently. A patentee may be his own lexicographer, but any special definition given to a word must be clearly defined in the specification or file history. Vitronics Corp. v. Conceptronic, Inc., 90 F.3d 1576, 1582 (Fed.Cir.1996).

*5 Additionally, a claim term should generally be read so as not to exclude the inventor's device or the inventor's preferred embodiment. *See, e.g., id.* at 1581 (claim interpretations excluding the preferred embodiment are heavily disfavored); Modine Mfg., 75 F.3d at 1550 ("[A] claim interpretation that would exclude the inventor's device is rarely the correct interpretation[.]").

While examining the patent specification is appropriate, it is improper to import, or "read in" to a claim, a limitation from the specification's general discussion, embodiments, and examples. *See, e.g.*, Intel Corp. v. United States Int'l Trade Comm'n, 946 F.2d 821, 836 (Fed.Cir.1991) ("Where a specification does not *require* a limitation, that limitation should not be read from the specification into the claims.") (internal quotation omitted); Constant v. Advanced Micro-Devices, Inc., 848 F.2d 1560, 1571 (Fed.Cir.1988) ("Although the specification may aid the court in interpreting the meaning of disputed language in the claims, particular embodiments and examples appearing in the specification will not generally be read into the claims.").

It is also improper to eliminate, ignore, or "read out" a claim limitation from a claim in order to extend a patent to subject matter disclosed, but not claimed. *See, e.g.*, Ethicon Endo-Surgery, Inc. v. United States Surgical Corp., 93 F.3d 1572, 1582-83 (Fed.Cir.1996) (court cannot read a limitation out of a claim); *see also* Unique Concepts, Inc. v. Brown, 939 F.2d 1558, 1562 (Fed.Cir.1991) (patentee cannot be allowed to expressly state throughout specification and claims that his invention includes a limitation and then be allowed to avoid that claim limitation in infringement suit by pointing to one part of specification stating an alternative lacking the specification).

Claims are not limited to the preferred embodiment. CVI/Beta Ventures, Inc. v. Tura LP, 112 F.3d 1146, 1158 (Fed.Cir.1997) ("as a general matter, the claims of a patent are not limited by preferred embodiments."); *see also* Amhil Enters., Ltd. v. Wawa, Inc., 81 F.3d 1554, 1559 (Fed.Cir.1996) ("A preferred embodiment ... is just that, and the scope of a patentee's claims is not necessarily or automatically limited to the preferred embodiment.").

Finally, when intrinsic evidence is unambiguous, it is improper for the court to rely on extrinsic evidence to contradict the meaning of the claims. *See* Pitney Bowes, Inc., v. Hewlett-Packard Co., 182

F.3d 1298, 1308-9 (Fed.Cir.1999). If, after considering the intrinsic evidence, a claim term is ambiguous, a court may look to extrinsic evidence to assist in determining the meaning or scope of terms in a claim. *Vitronics*, 90 F.3d at 1584. Extrinsic evidence includes expert testimony, inventor testimony, and technical treatises or articles. *Id.* Extrinsic evidence cannot, however, alter the clear meaning of a claim arising from the patent or prosecution history. *Id.*

DISCUSSION

I. Terms Proposed for Construction by Plaintiff

***6** Plaintiff seeks the construction of three claim terms or phrases: (1) automatically; (2) file; and (3) "processing by the third party forms servicer an electronic payment associated with the form."

A. Automatically

The term "automatically" appears in claims 1, 9, 12, 13, 14, 21, and 32 of the '278 patent, and in claims 4, 19, 36, and 38 of the '042 patent. In each instance, the term modifies one of three functions: populate, insert, or store. The term principally appears as "automatically" and occasionally as "automatic." This different use is immaterial to the construction of the term.

I previously construed the term in the December 19, 2002 Opinion in the *ApplyYourself* case. Dec. 19, 2002 Op. at pp. 10-29. Plaintiff's proposed construction is the same as the construction I adopted there. I agree with defendant that the constructions adopted in the *ApplyYourself* case are not controlling here. *Texas Instruments, Inc. v. Linear Techs. Corp.*, 182 F.Supp.2d 580, 586 (E.D.Tex.2002) (court's claim construction by another judge in same district in prior suit did not collaterally estop unrelated defendant from obtaining new claim construction; independent review of the claims would ensure fairness to all parties).

Nonetheless, while the previous claim constructions do not have preclusive effect here, to the extent neither party raises new arguments, I defer to the prior claim constructions. *KX Indus., L.P. v. PUR Water Purification Prods., Inc.*, 108 F.Supp.2d 380, 387 (D.Del.2000), *aff'd*, 2001 WL 902507 (Fed.Cir.2001). Additionally, even in the presence of new arguments, I give "considerable weight" to my previous claim constructions. *See Colby v. J.C. Penney Co.*, 811 F.2d 1119, 1123 (7th Cir.1987) (under the principles of *stare decisis* "a court must give considerable weight to [its own previous decisions] unless and until they have been overruled or undermined by the decisions of a higher court, or other supervening developments, such as a statutory overruling").

The construction of "automatically" that I adopted in the *ApplyYourself* case is: "Once initiated, the function is performed by a machine, without the need for manually performing the function." Dec. 19, 2002 Op. at p. 29. Defendant does not oppose this construction, but contends that because the term modifies different functions, the construction of the term must be made specifically in the context of the claim limitation in which it appears.

I disagree. "Automatically" describes how the function is to be performed. There is no suggestion from the claim language that the nature of that performance depends on the particular function being performed. Furthermore, generally, the same meaning is ascribed to the same claim term. *Omega Eng'g, Inc. v. Raytek Corp.*, 334 F.3d 1314, 1334 (Fed.Cir.2003) ("[W]e presume, unless otherwise compelled, that the same claim term in the same patent or related patents carries the same construed meaning."). Here, there is no compelling reason to construe "automatically" as it appears in each separate claim limitation. Thus, I recommend adoption of the prior construction of "automatically."

B. File

***7** This term appears in claims 30 and 32 of the '278 patent. I previously construed the term to mean "[a]n electronically stored collection of information that has a unique name." Dec. 19, 2002 Op. at pp. 30-32. Plaintiff contends that I should adopt the same construction here.

Defendant states that while it has no objection to this construction, the term should be construed in the context of the claim in which it appears. The term appears in the context of the phrase "application information file," in claim 32 of the '278 patent. But, it appears independently in claim 30 of the ' 278 patent, and it was separately construed in the *ApplyYourself* case. Given that it appears independently in at least one claim, it warrants its own independent construction, divorced from the "application information file" phrase. Accordingly, I recommend adoption of the prior construction.

C. "Processing by the Third Party Forms Servicer an Electronic Payment Associated With the Form"

This phrase appears in independent claims 1, 16, and 32 of the '042 patent. In its entirety, the claim phrase reads: "processing by the third party forms servicer an electronic payment associated with the form, the processed payment being from the form user to the one of the multiple institutions to which the form is directed[.]" 35:29-32; 36:45-48; 37:63-67. Both parties seek construction of this phrase, although defendant adds some additional claim phrases related to the "electronic payment function." For efficiency, I address all proposed constructions related to the electronic payment function, here.

The proper construction of this claim phrase was vigorously contested in the *ApplyYourself* case. After briefing and oral argument, I construed the phrase as follows:

> Using the received payment information to facilitate the clearance, settlement, and/or transfer of the electronic payment. The processing function includes, but is not limited to, processing by the business entity hosting the forms engine software and excludes any processing by any public form user or any of the institutions of higher education.

July 7, 2003 Op. at pp. 25, 45-55. Plaintiff argues for the adoption of this previous construction. Plaintiff additionally argues for the adoption of two terms within this claim phrase which I construed in the *ApplyYourself* case: (1) "electronic payment": "An electronic transfer of funds, such as an electronic check, credit card or debit card payment. The term electronic payment does not include a fee waiver." (2) "form": "A structured document having a collection of fields for entering and containing data. The form may be rendered to a user on a client computer or any web-browser enabled graphical display." *Id.* at p. 25.

Defendant does not dispute the prior construction of "electronic payment." In the *ApplyYourself* case, I concluded that the construction of "electronic payment" as "an electronic transfer of funds, such as an electronic check, credit card or debit card payment [and] does not include a fee waiver[,]" was appropriate because it was consistent with the claim language. July 7, 2003 Op. at p. 45. That same reasoning applies here. Accordingly, I recommend adherence to the prior construction for "electronic payment."

*8 Defendant proposes a different construction for "form": "an electronic document having fields for the entry and display of data, which consists of one or more pages." Curiously, while defendant offers this alternative construction, it makes no mention of the previous construction.

Defendant points to two parts of the '278 patent specification in support of its proposed construction of "form." First, defendant notes that the specification states that "[t]he present invention comprises a universal forms engine that permits the creation and processing of customizable electronic forms and selective sharing of information between the customized forms." 2:1- 4. Based on this, defendant argues that "form" means an electronic document. Next, defendant points to the specification's statement that "[a] form is considered to be essentially a container for data and implies an associated process." 2:22-23. Then, defendant notes that moreover, several claims of the '278 patent describe a form as having fields for the entry of information. 22:40-41, 22:57-58, 25:7-8, 26:11-13.

Next, because several of the dependent claims in the '278 patent refer to multiple form pages and multiple pages, 24:37-40, 25:53-54, 26:61, 26:64, defendant argues that the construction of "form" should include that the electronic document consists of one or more pages. Thus, defendant's proposal reads: "an electronic document having fields for the entry and display of data, which consists of one or more pages."

I recommend rejecting defendant's proposed construction of "form ." The word "electronic" is not necessary because first, it is redundant in that some uses of the word "form" in the patent claims and specification are preceded by "electronic." Second, my previous construction, by discussing how the form may be rendered to the user, implicitly defines "form" as being electronic.

Also, the construction does not need to expressly state that a form is one or more pages because by construing the word without reference to any page limitation, none is suggested. Furthermore, the characteristic of multiple pages is expressed in dependent claims. Cases hold that generally, limitations of dependent claims are not normally read into the independent claim from which they depend. Karlin Tech., Inc. v. Surgical Dynamics, Inc., 177 F.3d 968, 971-72 (Fed.Cir.1999).

Finally, I agree with plaintiff that there is no support for the inclusion of the word "display" in the construction of form. The specification provides that "a form is considered to be essentially a container for data and implies an associated process." 2:22-23. As such, plaintiff argues that a "display" requirement is not inherent in the meaning of the word form, nor is it required by the specification. I agree with plaintiff that while other claim language may suggest that the "form" is displayed, the term "form" itself does not. Therefore, I recommend rejecting defendant's proposed construction of "form" and adhering to the previous construction.

*9 As to the larger claim phrase at issue, defendant suggests that in addition to the claim phrase quoted above regarding electronic payments, other claim phrases related to the electronic payment function need to be construed. Defendant cites to the following claim language: "receiving by the third party forms servicer over the computer network ... electronic payment information entered by the user." This phrase appears in claims 1, 16, and 32 of the '042 patent. 35:26-28; 36:42-44; 37:60-62. Defendant also proposes that the slightly different language in claim 38 be construed: "receiving from the form user via the third party form servicer an electronic payment associated with the customized form[.]" 38:54-56.

Defendant offers separate constructions for five subparts of the originally construed phrase quoted at the beginning of this section and the additional phrases quoted in the preceding paragraph. The subparts proposed for construction are: (1) "processing ... an electronic payment associated with the form"; (2) "receiving ... electronic payment information"; (3) "by the third party forms servicer"; (4) "via the third party form servicer"; and (5) "entering payment information."

1. "By the Third Party Forms Servicer"

Because the central dispute in regard to this electronic payment phrase concerns the phrase "by third party forms servicer," I start with it. The heart of the dispute regarding this claim phrase in the *ApplyYourself* case was whether the entire function of processing the electronic payment had to be exclusively performed by the third party forms servicer itself or whether the third party forms servicer could contract with another party to perform the function. July 7, 2003 Op. & Ord. at pp. 44-55. I rejected the argument that the function had to be performed exclusively by the third party forms servicer. Thus, my claim construction indicates that the processing function includes processing by the business entity hosting the forms engine software, but that the processing is not limited to that entity. Defendant here contends that my prior construction was in error.

Defendant raises arguments similar to those raised by the defendant in *ApplyYourself* and which I considered and rejected. For example, defendant points to Figure 15 in the '042 patent as demonstrating that the payment processing occurs inside the forms engine operated by the third party forms servicer. But, as explained in the July 7, 2003 Opinion, this argument "fails to account for the fact that a fourth entity must be involved in the processing of an electronic payment because of the nature of all electronic payments." July 7, 2003 Op. at p. 53. Because the electronic payment function requires a financial intermediary which authenticates credit cards and verifies account balances and two banks, the function necessitates the involvement of entities other than the forms engine host. *Id.* at pp. 53-54. Even if the business entity hosting the forms engine were to acquire the ability to perform the credit card clearinghouse function, the electronic payment processing function still requires the participation of two banks whose functions cannot be delegated to a non-bank entity. *Id* . at p. 54. Thus, reliance on Figure 15 is unpersuasive.

***10** In the July 7, 2003 Opinion, I also addressed the defendant's argument that because the claim language provides that the third party forms servicer receives and processes both user information and electronic payment information and because the business entity hosting the forms engine is the third party forms servicer entity which processes the user information, it must be that same entity that processes the payment information. I rejected this argument in favor of plaintiff's argument that because these are "comprising claims," nothing in the claim language precludes another party from taking part in the processing of electronic payments. As I explained:

> However, plaintiff notes that these are "comprising" claims which recite required steps and elements, but which do not preclude additional steps or elements. Vehicular Techs. Corp. v. Titan Wheel Int'l, Inc., 212 F.3d 1377, 1382-83 (Fed.Cir.2000) ("The phrase 'consisting of' is a term of art in patent law signifying restriction and exclusion while, in contrast, the term 'comprising' indicates an open-ended construction.... In simple terms a drafter uses the phrase 'consisting of' to mean 'I claim what follows and nothing else.' A drafter uses the term 'comprising' to mean 'I claim at least what follows and potentially more." ') (citations omitted); Georgia-Pacific Corp. v. United States Gypsum Co., 195 F.3d 1322, 1327-28 (Fed.Cir.1999) (use of the word "comprising" means including the elements that follow, but not excluding additional, recited elements).
>
> Plaintiff argues that because these are "comprising" claims, nothing in the claim language precludes another party from taking part in the processing of electronic payments. Plaintiff contends that as "comprising" claims, the claims merely require that the third party forms servicer include the business entity hosting the forms engine as a party involved in the processing of payments, but not the sole party performing that function.
>
> I agree with plaintiff. Although the claim language noted by defendant requires the "third party forms servicer" to "receive" both user and payment information and then to "process" both user and payment information, it does not, by itself, limit the interpretation of "third party forms servicer" to the forms engine host business entity nor does it preclude that entity from utilizing a fourth party to participate in the "processing." I note that the parties themselves define "processing" to include "facilitation" of the electronic payment. This suggests that the role of the "third party forms servicer" is not restricted to the actual performance of the processing function, but may include a facilitator capacity.
>
> Given the nature of a "comprising" claim, additional elements may be part of the claim. Accordingly, based on the claim language, I interpret the disputed phrase "processing by the third party forms servicer," to mean that the processing function, as previously construed by the parties, includes, but is not limited to, processing by the business entity hosting the forms engine software and excludes any processing by any public form user or any of the institutions of higher education.

***11** July 7, 2003 Op. at pp. 48-49.

Defendant in the present case renews the argument made by the defendant in *ApplyYourself* and suggests that I misinterpreted the law regarding "comprising" claims. I disagree.

Defendant primarily relies on Moleculon Research

Corp. v. CBS, Inc., 793 F.2d 1261 (Fed.Cir.1986). There, the court rejected Moleculon's argument that "comprising" language opened the patent claim and its individual method steps to additional structural elements in addition to opening the claim to additional steps. *Id.* at 1271. The court concluded that Moleculon's position was too broad. *Id.* The court held that while

> a transitional term such as "comprising" or, as in the present case, "which comprises," does not exclude additional unrecited elements, or steps (in the case of a method claim), ... the transitional phrase does not, in the present case, affect the scope of the particular structure recited within the method claim's step.

Id.

Defendant also cites to a 1997 case for the proposition that " '[c]omprising' is a term of art used in claim language which means that the named elements are essential, but other elements may be added and still form a construct within the scope of the claim." Genentech, Inc. v. Chiron Corp., 112 F.3d 495, 501 (Fed.Cir.1997).

Defendant relies on these cases to argue that the term "comprising" cannot be used to read out the claim limitation's express requirement that processing an electronic payment be performed by a third party forms servicer. Defendant argues that the

> open-ended "comprising" term permits the inclusion of additional steps, which may or may not be performed by additional actors, but it cannot alter or abrogate the express requirement that the *third party forms servicer,* i.e., the same entity responsible for *processing* the *forms,* actually perform the *payment processing* step.

Deft's Op. Cl. Constr. Brief at p. 29.

What defendant fails to recognize is that the prior construction requires the business entity hosting the forms engine to retain responsibility for processing the electronic payment. The construction mandates processing by the business entity hosting the form (e.g. the third party forms servicer that also processes the user information) but allows some processing steps related to electronic payments to be performed by another entity, except the public form user or any of the institutions of higher education. Thus, the prior interpretation is consistent with the law regarding "comprising" claims because it keeps the "named element" of having the third party forms servicer that also processes the user information, process the electronic payment information, but it allows the extra step of a fourth entity participating in such processing along with that third party forms servicer. I fundamentally disagree with defendant's argument that all of the electronic payment function process must be performed by the host of the forms engine. That entity must perform some, but not all, of the electronic payment function. Accordingly, I recommend adhering to the prior construction for the reasons initially explained in the July 7, 2003 Opinion and expressed herein.

2. "Via the Third Party Forms Servicer"

***12** This phrase appears in claim 38 of the '042 patent. Defendant proposes the following construction: "[t]he institution taking possession of funds in its account through an electronic transfer of those funds from the form user, where the funds that are being transferred to the institution from the user have come by way of or by means of the third party form servicer." To the extent this phrase requires construction at all, I construe it consistently with my previous construction in the *ApplyYourself* case and consistently with the construction for "by the third party forms servicer" discussed in the preceding section. That is, the third party forms servicer, because it is required to be involved in the processing function as described above, may ultimately transfer the funds from the user to the institution, but other portions of the payment processing function may have been performed by a separate entity so long as it is not the user/applicant or the institution.

3. "Processing ... an Electronic Payment Associated with The Form"

Defendant proposes the following construction for this claim phrase: "[s]ubjecting an electronic payment associated with the form to[,] or handling an electronic payment associated with the form through[,] an established and routine set of procedures for effecting an electronic transfer of funds, including procedures for authorization, clearance and settlement."

Since "electronic payment" and "form" have already been addressed, the only remaining term in this phrase actually needing construction is "processing." Plaintiff proposes that "processing" in the context of the payment function be construed as "the manipulation of data within a computer system." Defendant argues that the term means " 'to subject to a special process or treatment (as in the course of manufacture" ' or " 'to subject to or handle through an established usu. routine set of procedures[.]" ' Deft's Op. Cl. Constr. Brief at pp. 25-26 (quoting Merriam Webster's Collegiate Dictionary 929 (10th ed.1994)). Thus, defendant's proposed construction for this claim phrase uses the terms "subjecting ... to[,] or handling ... through[,] an established and routine set of procedures[.]"

Defendant contends that because the claim specifies that the third party forms servicer processes an electronic *payment* rather than electronic *payment information* as recited in the previous limitation regarding the receipt by the third party forms servicer of electronic payment information entered by the user, the term "processing" as used in the phrase "processing by the third party forms servicer an electronic payment associated with the form," means something more than "processing" information or data. Defendant also contends that if "processing" means only the manipulation of data within a computer system, then the stated and claimed goal in the subsequent claim limitation which recites "relieving the institution of the administrative burden of processing forms and payments," would not be achieved.

***13** I disagree with defendant. First, the distinction between "electronic payment" and "electronic payment information" does not support defendant's construction. Under the claim language, an "electronic payment" gets "processed" while "electronic payment information" gets "received." Thus, there is no basis to conclude that the term "processing" when used with "electronic payment" means anything more than "the manipulation of data within a computer system." The previous function is restricted to receiving information which requires no manipulation of data.

Second, I reject defendant's argument that "manipulation of data within a computer system" is insufficient to relieve the institution of processing forms and payments. "Manipulation" is a broad term and is not confined, in this construction, to a narrow task.

Accordingly, I recommend that plaintiff's proposed construction for "processing" be adopted.

4. "Receiving by the Third Party Forms Servicer Over the Computer Network User Information and Electronic Information Entered by the User"

Defendant proposes the following construction: "[t]he third party forms servicer takes possession of the electronic payment information entered by the user, but need not do anything with it." Defendant notes that the act of "receiving" is passive and stands in contrast to "processing" which requires the third party forms servicer to do something.

I agree with defendant and conclude that this proposed construction is supported by the claim language. I also note that defendant's argument in regard to this phrase underscores my reasoning in regard to the construction of the term "processing." I recommend that defendant's proposed construction of the "receiving" phrase be adopted.

5. "Entering Payment Information Onto the Form"

Dependent claims 2 and 17 of the '042 patent provide a limitation in which the payment information entered by a user is entered onto the form. Defendant contends that the claims refer, as their antecedent basis, to the form claimed in independent claims 1 and 16, respectively. Thus, defendant argues, entering the payment information onto the form must mean "entering the payment information in some designated data field(s) of the form generated by the forms engine program and customized in its appearance and content in accordance with the preferences of the institution." I agree with defendant that this construction is a fair interpretation of the claim language and I recommend that it be adopted.

II. Terms Proposed for Construction by Defendant

Defendant groups its constructions into six different functions performed by the invention covered by the two patents. One of these groups addresses all of the claim limitations directed at the "electronic payment function." As noted above, the preceding discussion of plaintiff's proposed construction of "processing by the third party forms servicer an electronic payment associated with the form," includes a discussion of defendant's proposed constructions for the "electronic payment function" and there is no need to further address those constructions.

***14** The remaining five functions are: (1) forms engine; (2) user information database; (3) no rewriting of code; (4) forms processing; and (5) relief from administrative burden. In addition, defendant proposes constructions for two miscellaneous terms: (1) metadata; and (2) relational database.

A. Forms Engine Function

1. "Application Form" and "Application"

"Application" appears in claims 1 and 32 of the '278 patent and in claim 26 of the '042 patent. "Application form" appears in claims 1 and 32 of the '278 patent and in claim 39 of the '042 patent. Defendant proposes slightly different constructions for each term.

Defendant argues that "application form" should be construed as "a form representing an application for admission to a higher education institution." Defendant bases its proposal on the preamble to claim 1 of the '278 patent which recites: "[a] method of creating and processing over a computer network forms representing applications to different higher education institutions [.]" 22:34-36. Defendant argues that because the preamble indicates that claim 1 is directed to a method that results in application forms to "higher education institutions," the construction of "application form" must include a reference to such institutions.

In contrast, defendant proposes the following for the construction of "application": "[a] form representing an application to an institution." Defendant notes that because the preamble of claim 32 of the '278 patent refers only to "applications to institutions" and not "forms" for "applications" to "institutions of different higher education institutions," "application" must be construed more broadly to refer to all "institutions."

Plaintiff argues that neither of these terms need to be construed because they carry only their ordinary meaning and not a technical or special meaning. In such cases, construction is not necessary. Defendant suggests that the district court must construe every term proposed for construction by a party. Defendant's cited authority does not support this proposition. In *Sulzer Textil A.G. v. Picanol N.V.*, 358 F.3d 1356, 1366 (Fed.Cir.2004), the court held that "the district court must instruct the jury on the meanings to be attributed to all disputed terms used in the claims in suit so that the jury will be able to intelligently determine the questions presented." *Id.* (internal quotation omitted). That statement, however, was made in the context of resolving the question of whether a district court must instruct the jury on all the constructions it actually rendered. *Id.* at 1365-66. The court did not consider the question of whether a claim term which appears to be used in its ordinary sense, and not in any particular technical or scientific sense, must be construed simply because one party requests its construction.

While claim terms "must be construed as they would be understood by a person of ordinary skill in the art to which the invention pertains," and thus, "[w]hat the claim terms would mean to laymen is irrelevant[,]" *Searfoss v. Pioneer Consol. Corp.*, 374 F.3d 1142, 1149 (Fed.Cir.2004), if a person of ordinary skill in the art would understand the term in its ordinary, everyday sense, there is no need to construe the term. *E.g.*, Biotec Biologische Naturverpackungen GmbH & Co. KG v. Biocorp, Inc., 249 F.3d 1341, 1349 (Fed.Cir.2001) (district court did not err when it declined to construe "melting" when the meaning of "melting" did not depart from its ordinary meaning or otherwise require construction); Appelra Corp. v. MicrosMass, UK, Ltd., 186 F.Supp.2d 487, 524, 526 (D.Del.2002) (court declined to construe terms "maintain," "maintaining," and a "whereby" clause because they were clear on their face and the meaning was "self-evident"); Zip Dee, Inc. v. Dometic Corp., 63 F.Supp.2d 868, 872 (N.D.Ill.1998) (rejecting defendant's "artificial construct" of the term "tension" because no construction beyond the "ordinary English language meaning of the term" was required and thus, the patent's "references to 'tension' [would] go to the jury without the interposition of any judicial gloss.").

***15** Both "application" and "application form" are easily understood terms which the patents use in their ordinary sense. Neither the claim language nor the specification suggests that the meaning is anything other than the form used to apply to an institution or an institution of higher education. To the extent any construction is needed, I agree with plaintiff that it should be limited to "a form corresponding to an application."

I further conclude that only the term "application" needs the construction. "Application form" needs no further construction because it already clearly communicates its ordinary, everyday meaning in its own words. To apply the construction for "application" to "application form" would be an exercise in redundancy.

Additionally, there is no support in the claim language to restrict "application form" to an admissions application. The term "admissions" is not used to modify "application form" in claim 1 of the '278 patent. The specification of the '042 patent indicates that although the preferred embodiment of the invention is directed toward admissions forms, the invention may be used for "processing many different types of forms." 9:25.

Finally, I reject defendant's suggestion that "application form" is restricted to applications to "institutions of higher education" while "application" corresponds to the broader "institutions." The only reference to institutions of higher education is in the preamble to claim 1 of the '278 patent. It does not appear anywhere else in that claim and it does not appear in claim 32 of the '278 patent or in claim 39 of the '042 patent, claims which also refer to "application form." Generally, "[l]anguage in a preamble limits a claim where it breathes life and meaning into the claim, ... but not where it merely recites a purpose or intended use of the invention." *Innova/ Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1118 (Fed.Cir.2004) (citation omitted). In this case, the reference to "institutions of higher education" in the preamble is only a recitation of a purpose or intended use and adds no separate meaning to the claim.

Accordingly, I recommend that plaintiff's proposed construction of "a form corresponding to an application," be adopted for the term "application."

2. "Institution"

This term appears in claims 1, 21, and 32 of the '278 patent and in claims 1, 16, 32, and 38 of the '042 patent. Defendant proposes it be construed as "an established organization or corporation." Defendant's construction is based on a definition from Merriam Webster's Collegiate Dictionary 606 (10th ed.1994)..

Plaintiff argues that the term needs no construction because it is used only in its ordinary meaning. I agree with plaintiff that because the term is used only in its plain, customary meaning and there is no technical or scientific meaning ascribed to it, providing a construction for the term simply adds unnecessary complexity.

3. "Creating" or "Generating"

***16** One or both of these terms appear in claims 1, 2, 21, and 32 of the '278 patent, and in claim 1 of the '042 patent. Defendant again relies on Merriam Webster's Collegiate Dictionary to construe the terms as "bringing into existence." I agree with plaintiff that because these terms are ured only in their ordinary, everyday sense, there is no need to construe them.

4. "In Response to a Request"

This term appears in claims 1, 21, and 32 of the '278 patent and claims 1, 19, and 36 of the '042 patent. Defendant states that both "response" and "request" have their common, everyday meanings. Nonetheless, defendant proposes that the phrase be construed as "in reaction of an instance of a user asking for that form." Plaintiff argues that the phrase needs no construction as it is readily understandable without further elaboration. I agree with plaintiff that the phrase need not be construed because even as defendant notes, the words are used in their ordinary, plain meaning.

5. "Providing" and "Transmitting"

"Providing" is seen in claims 1 and 32 of the '278 patent and in claims 1, 16, 32, and 38 in the '042 patent. For example, in claim 1 of the '278 patent, it is used as follows: "providing to the applicant over a computer network the first application form[.]" 22:42-43. "Transmitting" appears in claims 6, 27, and 32 of the '278 patent. For example, in claim 32, it is used as follows: "transmitting the customized application over a computer network to a requesting applicant[.]" 26:17-18.

For "providing," defendant seeks the following construction: "making the generated form available to the user." For "transmitting," defendant proposes: "sending the generated form to the user." Defendant relies on Merriam Webster's Collegiate Dictionary for its proposed constructions.

Plaintiff argues that the terms "providing" and "transmitting" are everyday words used in their ordinary, everyday sense and thus, they need no construction. I agree with plaintiff. I further agree with plaintiff that neither the claim language nor the specification requires the construction to include the object of the action, e.g. "the generated form", or to whom it is directed, e.g. "the user." For example, if defendant's construction of "transmitting" were used, the claim phrase "transmitting the customized application over a computer network to a requesting applicant" in claim 32 of the '278 patent, would read: " 'sending the generated form to the user' the customized application over a computer network to a requesting applicant." I agree with plaintiff that this makes the claim limitation unreadable.

6. "Forms Engine Program" and "Forms Generator"

Claim 21 of the '278 patent and claim 1 of the '042 patent refer to the software program that generates a response to a request from a user. Claim 21 of the '278 patent provides: "a forms engine program operating on the server computer for generating a form from the form description information[.]" 25:3- 5. Claim 1 of the '042 patent states: "... the form being generated by a forms generator that generates multiple forms corresponding to multiple institutions of higher education, the forms generator generating forms that are ..." 35:11-14.

***17** Defendant proposes one construction for both phrases: "a software program responsible for performing, among other tasks, the creation or generation of multiple forms corresponding to multiple institutions." Plaintiff proposes separate, but simpler phrases: "A software program that can be used to generate a form" for the term "forms engine program" and "a software program that can be used to generate forms" for the term "forms generator."

Both parties' proposals incorporate the phrase "a software program." I agree with the parties that the phrase "forms engine" is used in the technical, computing sense to mean "a software program." I further agree, as is seen in both parties' proposals, that the phrase "forms engine program" can generally be understood as a software program that creates or generates forms.

Plaintiff contends that defendant's proposal inappropriately inserts "among other tasks" which suggests that the forms engine program and forms generator are required to perform unspecified other tasks, other than generating forms. I agree with plaintiff that the claim language and specification does not support a construction suggesting that the forms engine program or forms generator must be able to perform other tasks. While the forms engine program or forms generator may actually be able to do so, the claims do not mandate the performance of other tasks. The disclosed function for forms engine program and forms generator appears limited to generating forms.

I agree with defendant that the construction for both terms properly includes the reference to "multiple forms corresponding to multiple institutions." Plaintiff notes that in claim 21, the term "forms engine program" is used only in conjunction with the generation of a form, in the singular, not multiple forms. Dependent claim 23 in the '278 patent discusses a forms engine program that can generate more than one form. But, plaintiff argues, the requirement that it generate more than one form does not derive from the phrase "forms engine program" itself, but from additional language in the claim.

I disagree with plaintiff. The preamble to claim 21 provides for "[a] system for creating an processing customized forms for unrelated institutions." 24:52-53. This establishes that the purpose of the claim is to create more than one form. Each step in the claim discusses "form" in the singular because the system generates only one form at a time. But, the invention, to fulfill its purpose, must include a forms generator or forms engine program that generates multiple forms. The whole point of the invention is that a single forms generator or forms engine program can generate forms for multiple institutions and populate subsequent forms with data stored from earlier forms. A construction of "forms generator" and "forms engine program" without the requirement of generating multiple forms for multiple institutions would exclude the invention.

***18** This distinguishes this reference to the preamble from the one discussed above in connection with the terms "application" and "application form." There, the preamble's reference to "institutions of higher education" was not a separate claim limitation because it merely recited a purpose of the invention. Here, while the use of the plural "forms" also indicates a purpose of the forms engine program, the plural term is required for the forms generator and forms engine program to have any meaning which comports with the invention.

Notably, defendant's proposed construction does not require that the forms engine program or the forms generator generate multiple forms simultaneously. The requirement is only that the forms engine program or forms generator be able to generate more than one form, not that it do so at the same time.

Thus, I recommend that the terms "forms engine program" and "forms generator" both initially be construed to mean "a software program which creates or generates multiple forms corresponding to multiple institutions."

Finally, although perhaps not obvious in the proposed constructions of these phrases, the briefing reveals that the parties dispute whether "a software program" as used in the construction of "forms generator" and "forms engine program" is a single program or multiple programs. Defendant contends that the "forms engine program" or "forms generator" is a single program that generates multiple forms corresponding to multiple institutions. Defendant relies on a reference in claim 1 of the '042 patent to "*a* forms generator" in the singular, "that generates multiple forms ..." 35:12-14 (emphasis added). Defendant also notes that the prosecution history reveals that one of the named inventors differentiated the invention disclosed in the '278 patent from an earlier version of the system called "ApplyWeb I," by noting that ApplyWeb I required a separate software program for each application form for each school. Exh. A to Deft's Op. Cl. Constr. Brief. [FN3] Thus, defendant contends, the invention disclosed in the patents in suit must be to a single program.

FN3. I request that in the future, defendant paginate all exhibits and refer to specific pages of an exhibit when citing to it.

Plaintiff counters this argument by noting that the word "a", as used in defendant's proposed construction "a software program" is typically understood to mean "one or more" in patent claims. *Tate Access Floors, Inc. v. Interface Architectural Resources, Inc.*, 279 F.3d 1357, 1370 (Fed.Cir.2002). Thus, by using "a software program," the construction means one or more programs. Plaintiff also notes that it is common for a "program" to include other "code," which may also be considered a "program," or to call upon other "programs," containing certain functions, with the "programs" working together to create a desired result.

Plaintiff argues that as long as the "forms engine program" or "forms generator" is involved in the generation process, the claim language is satisfied. Plaintiff also contends that nothing in the plain language of the claim deviates from this typical understanding or requires the "forms engine program" or "forms generator" to be the only software included in the form generation.

***19** Plaintiff argues that as long as the "forms engine program" or "forms generator" is involved in the generation process, the claim language is satisfied. Plaintiff also contends that nothing in the plain language of the claim deviates from this typical understanding or requires the "forms engine program" or "forms generator" to be the only software included in the form generation.

Furthermore, plaintiff argues, the intrinsic evidence contradicts defendant's position. In the preferred embodiment, the "forms engine" operates in concert with other software such as the web server software and the database management system software. Plaintiff argues that while the specification states that the "preferred implementation of the invention comprises a single forms engine program ...", this statement is limited to the preferred implementation and implies that implementations using more than one forms engine are possible.

I agree with plaintiff. Because the forms generator or forms engine program may use other code or programs to actually generate the form, and because it generates the form only in tandem with the web server and the database management software, it cannot be restricted to a single program. While it could be just a single software program that generates the forms, it should not be confined to a single program. Thus, I recommend that the following construction be adopted for "forms generator" and "forms engine program": "a software program, which, with or without additional software programs, creates or generates multiple forms corresponding to multiple institutions."

7. "Form Description Information," "Application Description Information," and "Application Information File"

While defendant proposes different constructions for these three terms, I consider them together because they are related. "Form description information" appears in claim 21 of the '278 patent as follows: "first data storage in communication with the server computer and including form description information specifying the content and appearance of each customized form [.]" 24:59-62. The phrase appears again in that claim: "a forms engine program operating on the server computer for generating a form from the form description information in response to a request for the form transmitted ..." 25:3-5.

"Application description information" appears in dependent claim 2 of the '278 patent: "[t]he method of claim 1 in which creating a first application form customized in accordance with the preferences of the first institution includes generating a first application in accordance with stored application description information ..." 23:16-20.

Defendant proposes parallel constructions for these two terms. First, for the "form description information" phrase, defendant proposes the following construction: "information describing a form that is sufficient to enable the forms engine program to generate the described form." For "application description information," defendant proposes: "information describing an application form that is sufficient to enable the forms engine program to generate the described application form." Defendant notes that the '278 patent specification uses "form information" and "application information" interchangeably to describe the information stored in the application data file. 5:61-63; 6:19-22. Accordingly, defendant contends that the two phrases should be similarly construed.

***20** Plaintiff does not disagree that "form description information" and "application description information" should receive parallel constructions. But, plaintiff contends that there is no reason not to adopt the construction I gave "form description information" in the *ApplyYourself* case and then use that as the basis for the parallel construction of "application description information."

The construction I rendered in the *ApplyYourself* case for "form description information" was "the information used to customize a form." July 7, 2003 Op. at pp. 38-43. My analysis of the meaning of the term rendered in the *ApplyYourself* case is equally applicable here. The construction adopted there is consistent with the ordinary meanings of the terms in the phrase. Additionally, I am reluctant to adopt a construction that incorporates another construed phrase as defendant proposes here. Defendant's proposed construction uses "forms engine program." Thus, the jury will have to cross-reference the construction for "forms engine program" to understand the meaning of "form description information." Because the construction from the *ApplyYourself* case sufficiently explains the phrase while avoiding this cross-referencing problem, I recommend that that construction be adopted in this case for the term "form description information." I also recommend that "application description information" be construed to mean "the information used to customize an application."

The phrase "application information file" appears in the following step of claim 32 of the '278 patent: "providing at least two application information files, each describing a customized application for an institution[.]" 25:64- 65. It also appears again in a following step: "generating a customized application in response to a request over a computer network from an applicant, the application form and content being specified by one of the at least two application information files, ..." 26:8-11.

In the *ApplyYourself* case, I construed "application information file" as "a file that stores information that includes a description of a distinct application form. The file describes the form itself, not the user data (e.g. student specific information) that may ultimately be entered into a particular copy of the form." Dec. 19, 2002 Op. at p. 30.

Defendant concedes that the previous construction is generally correct, but defendant argues that more specificity is required. Without citing to any part of the patent specification, defendant contends that both the specification and the claim language suggest that "application information file" should be construed to mean "a uniquely named text or template file that contains the instructions and pattern descriptions that enables the forms engine program to create a distinct application form that is customized in its appearance and content."

Plaintiff contends that defendant's proposal introduces new phrases without any support, such as "pattern description," that are confusing and undefined. As such, and because defendant concedes that it does not disagree with our prior construction, plaintiff argues that I should adopt my prior construction.

***21** I agree with plaintiff. Defendant's proposal unnecessarily adds new, undefined phrases which will only lead to increased complexity in the claims construction process.

Furthermore, with a separate definition of "file" as rendered above, ("an electronically stored collection of information that has a unique name"), there is no need to construe "application information file" as something "uniquely named." I do agree with defendant that the addition of the words "text or template" to modify "file" is warranted by the claim language. In the discussion of the construction of "application information file" in the December 2002 Claims Construction Opinion on the '278 Patent. I noted that the use of the word "file" in that claim phrase was a "text, or perhaps template, file that stores the directions to produce the customized form for each institution." Dec. 19, 2002 Op. at p. 31. While this reference to "text, or perhaps template," did not make it into the final claim construction of the phrase, I conclude that the term "application information file" should be construed with that modification of "file." Thus, I recommend that the following construction of "application information file" be adopted: "A text or template file that stores information that includes a description of a distinct application form. The file describes the form itself, not user data (e.g. student specific information) that may ultimately be entered into a particular copy of the form."

B. User Information Database Function

This function is initially seen in the following language from the three independent claims of the '278 patent:

> storing the posted applicant information in a database having a database field structure defined by multiple database fields, the database including multiple records, each record capable of storing information corresponding to each of the database fields[.]

22:49-52; 24:55-67; 26:4-7.

The following claim language from claim 1 of the '278 patent also encompasses the "user information database function":

> automatically storing the applicant information entered into the second form data fields into the database by adding new records to the database, the automatic storing of the applicant information not altering the database field structure, thereby allowing new form data fields corresponding to applicant information not previously requested to be added to an application form without requiring alterations of existing application forms or of programs that access the database, whereby customized applications to different institutions share data through common, extensible data storage.

23:5-15. The other independent claims express a similar function. 25:16-23; 26:25-33.

Defendant also points to dependent claim 11 and this particular language as being relevant to the user information database function:

> The method of claim 1 in which storing the posted applicant information in a database having a database field structure defined by multiple database fields includes parsing the applicant information within a[sic] into data elements, the data elements being separately stored and identified, thereby allowing the data elements to be separately retrieved and rearranged in subsequent applications.

*22 23:66-67--24:1-5.

From these claim limitations, defendant proposes constructions for: (1) database; (2) database field structure; (3) defined by multiple database fields; (4) multiple records; (5) record; (6) data element; (7) by adding new records to the database without altering the database field structure; and (8) extensible.

The claim language at issue here, unlike other words or terms in the claims, is not used in its ordinary, customary sense, and is used in a technical sense. Thus, construction is required.

In regard to this function, one of the fundamental issues is whether the database storage is exclusively in a format, or structure, that is based on the concept of tables as one ordinarily thinks of a table for organizing information, a combination of rows and columns. While a table-based structure appears to be what is expressed in the preferred embodiment, *see* 3:48 (noting that the preferred embodiment uses "relational databases" (discussed below in section entitled "Miscellaneous Terms"); 9:13-14 (noting use of "transactions database table" and "transactions operations table" in preferred embodiment); 9:28 (section describing "Attribute Table"); 9:67 (section describing "User Attribute Sent Table), the specification also expressly states that the "invention is not limited ... to ... the use of any particular ... database," 3:49-51, and it further discloses the use of Extensible Markup Language

(XML) as an alternative method of storing user information. 21:13-67--22:1-19.

Accordingly, while the following discussion examines the claim terms at issue in this "user information database" function in the context of the preferred embodiment, I recognize that the claim terms are not limited to that embodiment both by the express statements in the written description and under general precepts of claim construction law. *CVI/Beta Ventures,* 112 F.3d at 1158 ("the claims of a patent are not limited by preferred embodiments.").

1. "Database"

Defendant proposes the following construction: "an organized collection of information that can be searched, retrieved, changed, and sorted using a collection of programs known as a database management system." Plaintiff proposes: "an organized collection of information that can be searched, retrieved, changed, and sorted using software." The point of contention is in whether the database uses "a collection of programs known as a database management system" or uses "software."

Defendant's proposal is the definition of "database" given in the 1995 edition of the *Dictionary of Computer Words* 61 (1995 rev. ed.) (relevant page found in Exh. B to Deft's Op. Cl. Constr. Brief). Defendant contends that the '278 patent specification does not indicate that the term "database" is used in any way other than this ordinary technical meaning.

Plaintiff argues against defendant's "database management system" limitation as unduly narrow. Plaintiff notes that the specification states that information can be stored in "tables" or in "XML files." 9:29-10:40, 21:14-19. Plaintiff further notes that while a database management system is often associated with accessing data stored in "tables," persons in the field and skilled in the art might not refer to the software that works with XML files as a "database management system." Accordingly, plaintiff argues, the claim language should not be limited to a "database management system."

***23** I agree with plaintiff. From the written specification, the parties' briefing, the expert declarations, and the arguments presented in the case, it appears that one skilled in the art of database systems would initially understand the ordinary, customary use of the term "database" to refer to tables. But, the written specification, the briefing, the expert declarations, and the arguments presented in the case also show that XML is at least one other "format" available in which to store data. To avoid limiting the term "database" to the concept of storage of data in tables, I recommend rejecting defendant's proposed construction with its inclusion of "database management system" which one of ordinary skill in the art could use to infer that the database at issue in the patent is restricted to one using tables. Thus, I recommend that "database" be construed as "an organized collection of information that can be searched, retrieved, changed, and sorted using software."

2. "Database Field Structure"

Defendant proposes this construction: "the structure of database fields, i.e. relations and the attributes or fields that define the columns the relations contain." Plaintiff proposes: "the grouping and organization of database fields."

To understand defendant's proposed construction, it is necessary to define some of the terms defendant uses. According to one authority, a "relation" is "a two-dimensional table in which data are arranged." Hector Garcia-Molina, et al., *Database Systems--The Complete Book* 61-62 (2002) (relevant page found in Exh. B to Deft's Op. Brief). An "attribute" is a name describing the meaning of an entry in a column of a relation. *Id.* at p. 62 (showing diagram of two-dimensional table with headings for four columns and noting that the "attribute describes the meaning of entries in the column below."). In the context of the patents in suit, an attribute in a two-dimensional table could be something like "first name," or "street address" or "user identification."

A "field" is the portion of the database that stores a data value for a particular attribute. *See* Pltf's Initial

Cl. Constr. Brief at pp. 9-10. Another definition for field is "a space reserved for a specified piece of information in a data record." Bryan Pfaffenberger, Ph.D., *Que's Computer & Internet Dictionary* 133 (6th ed.1995) (relevant page found in Exh. B. to Deft's Op. Cl. Constr. Brief). In this sense, a "field" refers to the location in the database in which a particular type of data is stored. *See* Pltf's Exh. 1 to Sept. 9, 2004 Oral Arg. at p. 4 (Claim Construction Statement showing construction of "field" as "a location in a record in which a particular type of data is stored. For example, EMPLOYEE-RECORD might contain fields to store Last-Name, First-Name, etc.").

With these definitions, defendant's proposed construction can be read as: "the structure of database fields, i.e., tables and the attributes or spaces that define the columns the tables contain." One of the problems with defendant's proposed construction is its reliance on technical terms to define the claim phrase "database field structure." Defendant's proposal requires several additional definitions or interpretations to be understood, unnecessarily complicating the claim construction.

***24** The more fundamental problem, however, is that once the technical terms used by defendant are defined, it is obvious that defendant's proposed construction limits the database field structure to a structure based on tables. As explained above, the patent's preferred embodiment of "database" may be tables, but it is error to so limit it.

The meaning of "database field structure" is not apparent from the claims themselves. To the extent the claims themselves give some definition to the term, it is limited to the modifying phrase immediately following "database field structure" which reads "defined by multiple database fields[.]" Thus, the claims disclose only that the "database field structure," whatever it is, must have "multiple database fields." Defendant represents that there is no mention in any part of the specification of "database field structure." Plaintiff does not dispute this representation and my independent review of the specification has revealed no reference to the exact term. The only relevant specification reference I found was to the following similar phrase: "As described in more detail below, information about the applicants is maintained as a set of attributes, each attribute corresponding to database fields." 7:29-31.

Given the lack of information in the claims themselves and in the specification, defendant relies on testimony from its expert Jeffrey Ullman, Ph.D., who explains that although "database field structure" is not a term that would be readily recognized by one of ordinary skill in the art of database systems, such a person would understand the phrase to refer to a "specification of the fields used in some single relation or file of records." Aug. 15, 2004 Ullman Declr. at ¶ 7. By referring to "relation," Ullman's explanation, which provides the foundation for defendant's proposed construction, inappropriately restricts the definition of "database field structure" to tables.

Consequently, I recommend that the phrase "database field structure" be interpreted to mean "the grouping and organization of database fields" with the understanding that "database field" refers to "the space reserved in the database for storage of a particular type of data."

3. "Defined by Multiple Database Fields"

Defendant proposes that this phrase be interpreted as "a set of attributes of a single relation intended to hold information about the applicants or users, as the case may be." As indicated above, the phrase "defined by multiple database fields," modifies it predecessor phrase "database field structure." Defendant construes "database fields" as the attributes of a single two-dimensional table. Based on this reasoning, defendant contends that one skilled in the art would understand the phrase "defined by multiple database fields" to refer to the structure or schema of a table and not to the structure or schema of the database itself. Defendant uses "applicants or users" because claims 1 and 32 refer to the storage of application information and claim 12 refers to the storage of user information.

***25** As noted above, while the specification reveals the use of a table-based database, it does not limit the type of database to one using a single table or multiple tables. As seen in plaintiff's September 9, 2004 oral argument presentation, there are several different database structures familiar to those skilled in the art. Pltf's Exh. 2 to Sept. 9, 2004 Or. Arg. at pp. 21-23. The patent claim language and specification disclose an invention which may employ any number of database structures to satisfy the "storing" claim limitation.

Defendant's proposal would limit the claim language to the preferred embodiment which describes the use of a three-column database structure headed by fields for "user identification," "attribute identification," and "data value." *Id.* at p. 24; *see also* Pltf's Exh. 5 to Oct. 6, 2004 Or. Arg. at pp. 9, 15 (showing preferred embodiment as single two-dimensional table with columns for "applicant identifier," "characteristic identifier," and "value"). For the reasons discussed above, a construction limiting the term to the preferred embodiment is unduly narrow and is contradicted by the specification's express disclosure that a table-based database is not the only method of storing user information data.

Thus, I recommend construing "defined by multiple database fields" as "multiple spaces for the storage of multiple types of data." Accordingly, the entire claim phrase "database field structure defined by multiple database fields" would mean "the grouping and organization of multiple spaces in the database reserved for the storage of particular types of data."

4. "Multiple Records and Each Record Capable of Storing Information Corresponding to Each of the Database Fields"

Defendant argues that "multiple records" means "the rows of a single relation" and that "record" in the phrase "each record capable of storing information corresponding to each of the database fields," means "a complete unit of related data items stored in named data fields." Defendant's proposals are based on the following definition of "record":

> In a database management program, a complete unit of related data items stored in named data fields. In a database, data record is synonymous with row.
>
> A data record contains all the information related to the item the database is tracking. Most programs display data records in two ways: as data-entry forms and as data tables. In a table-oriented relational database management system, the data records are displayed as horizontal rows and each data field is a column.

Que's Computer & Internet Dictionary 124 (found in Exh. B to Deft's Op. Brief).

While it may not be readily apparent from defendant's proposed construction of "record," the underpinning of defendant's interpretation is that a "record" is limited to one "row" of a single two-dimensional table. Thus, "multiple records" means multiple rows of such a table.

***26** In contrast, plaintiff construes "record" as a "collection of related data treated as a unit." Plaintiff explains that in the context of online admissions applications, a record might consist of all the data for a particular applicant's application such as name, address, high school attended, etc. Plaintiff contends that this application data may be organized in several different ways in a database with each carrying a different concept of "record":

(1) as multiple tables, with different tables storing different parts of the data, all tables being linked together by the applicant's identification number or some other linking principle. In that case, the "record" is the collection of linked data. *See* Pltf's Exh. 5 to Oct. 6, 2004 Or. Arg. at p. 11.

(2) all data for a single applicant are stored in a single row of a single table. In that case, the "record" is the data on that row. *Id.* at p. 7.

(3) as multiple rows of a single table, in which case the rows, together, would constitute the "record." *Id.* at pp. 9, 15.

(4) stored as a related set of XML data, in which case a record is the set of values that are linked to a "document."

Because, plaintiff's argument goes, some of the various database structures contemplated by the patent carry a meaning of "record" that encompasses more than one "row," or, in the case of XML data, no "row" at all, defendant's proposed construction of "multiple records," which is premised on its definition of "record" as a single row of a two-dimensional table, must be rejected. And, plaintiff continues, its proposal is superior because it captures all of the possible concepts of "record" suggested by the various database structures plaintiff describes.

Plaintiff contends that its invention encompasses all of the database structures it describes and that its construction of "record" is broad enough to take on different meanings of "record" in different steps of the claims. For example, in a multiple table model, id. at p. 11, or a model expressed by the preferred embodiment where there is one table with three columns and each row contains a user identification, an attribute identifier, and a value, the "record," according to plaintiff, is a collection of all of the rows containing information about a single applicant. Because each row in the multiple table model or in the three-column structure described in the preferred embodiment contains information about one of the applicant's attributes, the "record" should be thought of as all of those rows put together.

In support of this concept, plaintiff cites to a particular part of the specification. In describing a "User Attribute Sent Table," the specification refers to the previously described "User Attribute Table," which stores the values assigned to attributes for individual applicants. 9:45-46. The "User Attribute Table" is configured, in the preferred embodiment, as a single table with three columns, one for user identification, one for attribute identification number, and one for data value. [FN4] 9:45-48. Each row of the table contains the information related to one attribute for one applicant.

> FN4. The preferred embodiment actually discloses four columns with the additional column for attribute identification number sequence which would be used to assign a relative sequence to the attributes. Because the visual aides shown by the parties omit that fourth column, I do not use it here.

*27 The "User Attribute Sent Table," rather than storing the user information by attribute, stores the information contained in a completed application as a "snapshot of the completed application." 10:1-5. The specification further provides that:

> [t]he structure of the User Attribute Sent Table is very similar to that of the User Attribute Table. The primary key of the User Attribute Table is a user identifier (the users log-on name), whereas the primary key of the User Attribute Sent Table is a Transaction Identifier, which identifies a unique combination of user, application, and application terms. *Thus, there can be multiple records for a single user in the User Attribute Sent Table if the user has submitted multiple applications or the same application for different application terms.*

10:5-14 (emphasis added). I understand plaintiff's argument to be that (1) this description of the "User Attribute Sent Table" suggests that multiple records equates with multiple applications; (2) one application contains more than one attribute; (3) the structure of the User Attribute Table requires multiple rows for multiple attributes; (4) the User Attribute Sent Table, because it is the same or similar to the User Attribute Table, would also store its attributes on multiple rows of the table; and (5) therefore, the User Attribute Table's reference to "multiple records" implies that "record" consists of the collection of information from several rows related to one applicant.

I disagree with plaintiff that this portion of the specification supports its construction of "record" as all of the information related to one applicant. Rather, although the specification indicates that the User Attribute Sent Table uses a similar structure to that described in connection with the User Attribute Table, meaning a single two-dimensional table with columns and rows, it does not suggest that each "record" comprises multiple rows of information related to one applicant. In the User Attribute Sent

Table, it appears possible that a "record" is a single row. Given that the User Attribute Sent Table apparently stores user information as it appears in a single application, each row could include just the transaction identifier and all of the information contained in one application. Thus, one row is one completed application containing all of the user attributes sent. As such, there would be "multiple records" for "multiple applications" with "reccrd" referring to one "row." Thus, the quoted portion of the specification does not confirm that the proper interpretation of "record" in the context of the storing step, means all of the information from all rows relating to one applicant.

The other problem with plaintiff's argument is that to satisfy additional claim limitations, plaintiff must offer a different interpretation of "record." This is contrary to claim construction standards which ordinarily require the same term in a claim to be interpreted consistently. Omega Engineering, 334 F.3d at 1334.

***28** The automatically storing step provides, in relevant part, that the invention

> automatically stor[es] the applicant information entered into the second form data fields into the database by adding new records to the database, the automatic storing of the applicant information not altering the database field structure ...

23:5-9; *see also* 25:16-10; 26:25-29. Because this step addresses an applicant's second application, the applicant already has user information stored in the database. If "record" is defined as all of the information pertaining to one applicant, it does not fit within this claim step because the invention would not be adding a "new record" for the applicant. An old record, e.g. all of the information pertaining to one applicant, already exists. The only interpretation of "record" that satisfies this claim limitation is one that considers "record" to be a single "row," at least in the table-based data storage model.

Given this problem, plaintiff argued at the October 6, 2004 oral argument, that "record" must be interpreted in two different ways in claim 1 of the '278 patent. It must initially be considered as all of the information pertaining to one applicant for the first storing step, but then only as a single row for the automatically storing step. I recognize that plaintiff's proposed construction for "record" is broad enough to encompass both meanings because different groupings of "related data" can each be thought of as a "unit," but I cannot accept that even within the preferred embodiment's three-column two dimensional table, plaintiff must rely on two different meanings of the term "record." I conclude that this is inherently inconsistent with basic precepts of claim construction law.

On the other hand, defendant's premise that a "record" is a single row works with all steps of the preferred embodiment, as well as with a structure using multiple tables, and with XML. With the preferred embodiment, a "record," when considered a "row," is consistent with the use of "record" in both the first storing step as well as the automatic storing step which refers to the information from the second application.

In the storing step, the database includes "multiple records, each record capable of storing information corresponding to each of the database fields." In the three-column table expressed by the preferred embodiment, there are multiple rows, and thus, multiple records, and each row is capable of storing information corresponding to each of the database fields.

In the automatic storing step, the invention stores the applicant information entered in the second application's data fields in the database by adding new records to the database. With record meaning row, this limitation is easily satisfied by adding new rows to the database. This makes sense in that in the preferred embodiment, each attribute receives its own row in the three-column structure. With the first application storage of information expressed in the storage step, the applicant will have several rows stored in a table (again, this is in the context of the preferred embodiment), each row corresponding to a particular attribute. With record meaning row, the new attributes from the second application, e.g. the attributes not part of the first application, will be stored as a new record, that is, a new row.

*29 Because defendant's proposal for "multiple records" ("the rows of a single relation") implies that a record is a row in the context of a table-based database structure, and such a construction is consistent with the use of "record" in the storing claim limitation as well as the use of the "record" in the automatic storing claim limitation, in the context of the preferred embodiment, I recommend concluding that in the preferred embodiment, record should be understood as a single row in the table.

I reject, however, defendant's actual proposal for "multiple records" as "the rows of a single relation" because, again, this proposal restricts the term to the two-dimensional table expressed in the preferred embodiment which is inconsistent with the specification and claim construction standards. Rather, I recommend construing "record" to be "a collection of related data items stored in named data fields" and "multiple records" to mean "multiple collections of related data items stored in named data fields." In the preferred embodiment, this construction includes the understanding that one record is one row. In other embodiments, however, the restriction of record to one row may not be workable. For example, with XML, there is no traditional "row."

As seen in plaintiff's hearing exhibits, there are data items and data fields in a database structure using XML. Pltf's Exh. 5 to Oct. 6, 2004 Or. Arg. at p. 14. The data fields are any of the spaces where information will be stored, such as the spaces between the < > symbols or the space between the > < symbols. *Id.* The data items are, for example "<username>" and "872." *Id.* A collection of related data items stored in named data fields could be "<username>" and "872." That would comprise the record in the XML system.

This understanding meets both the storing and automatically storing claim limitations. Each record is capable of storing information related to each of the database fields and a new record is added to the database from the second form data fields. Accordingly, I recommend that "record" be construed as "a collection of related data items stored in named data fields."

5. "Data Element"

Dependent claim 11 of the '278 patent provides for

> [t]he method of claim 1 in which storing the posted applicant information in a database having a database field structure defined by multiple database fields includes parsing the applicant information ... into data elements, the data elements being separately stored and identified, thereby allowing the data elements to be separately retrieved and rearranged in subsequent applications.

23:66-67--24:1-5. Defendant proposes the following construction of "data element": "the smallest, indivisible unit of data stored in the database, which in the context of a relation, is a single component of a row, corresponding to a particular attribute."

Defendant contends that based on the specification, "data element" should be understood to refer to the smallest, indivisible unit of data stored in the database. The specification notes that "[t]o avoid having applicants enter data more than once to accommodate changes in format, the information is preferably stored in simpler data elements, and then combined during second stage validation into the format requested by the institution." 15:36-40. Additionally, each "data element" maps to a unique attribute having "a unique identifier or alias." 7:39-49. Defendant argues that in the field of database design, one would understand a "data element" to mean a single component of a row, corresponding to a particular attribute.

*30 Plaintiff proposes the following construction: "the smallest unit of data defined for use by a system. By way of example, a form may provide a single field for 'full name,' which can be defined to contain the data elements 'first name,' 'middle name,' and 'last name.'" ' I recommend the adoption of plaintiff's proposal. Defendant's proposal expresses the construction in the context of the preferred embodiment two-dimensional table. Plaintiff's proposal is more easily adapted to other database structures.

6. "By Adding New Records to the Database Without Altering the Database Field Structure"

As stated above, this is part of the "automatic storing" function related to information obtained from the second application, seen in independent claims 1, 21, and 32 of the '278 patent. 23:5-9; 25:16-20; 26:25-29. Defendant proposes that this claim language be construed to mean "the addition of new records, or rows, to a relation does not alter the structure or schema of the relation."

I recommend that this claim phrase not be further construed. First, defendant again ties its proposal to a two-dimensional database structure. For the reasons previously explained, this is inappropriate. Second, I have already construed the individual terms "record," "database," and "database field structure." The only additional words are "adding," "new," "without," and "altering," none of which are used in any sense but their ordinary, customary meaning. Accordingly, there is no need to further construe this phrase.

7. "Extensible"

Defendant also proposes to construe the word extensible which appears at the end of the "automatic storing" function. The entire claim phrase reads:

> automatically storing the applicant information entered into the second form data fields into the database by adding new records to the database, the automatic storing of the applicant information not altering the database field structure, thereby allowing new form data fields corresponding to applicant information not previously requested to be added to an application form without requiring alterations of existing application forms or of programs that access the database, whereby customized applications to different institutions share data through common, extensible data storage.

23:5-14; *see also* 25:16-23; 26:25-33.

Defendant proposes that "extensible" be construed as having the ordinary and customary meaning of "capable of being extended." Defendant contends this is supported by the specification. 7:31-37 ("If an institution chooses to include in its application a request for an applicant attribute that does not correspond to one included in the database, the database is easily extended to include the new applicant attributes without reprogramming the forms engine."). Defendant contends that construction is necessary, despite the term possessing its ordinary and customary meaning, to make clear that the term does not refer to a technical meaning of "extensible" in the field of database design. As defendant explains, at least one technical Internet dictionary refers to an "extensible database" as a database management system that allows access to data from remote sources as if the remote data were part of that database. Deft's Op. Cl. Constr. Brief at p. 17 n. 11 (citing to www.hyperdictionary.com).

***31** I agree with defendant. As explained above, ordinarily, when a claim term is used only in its common, customary sense with no particular technical or scientific meaning, it is not necessary to construe the claim. However, here, to prevent the jury from mistakenly assuming, or the parties arguing, that in this context "extensible" refers to a technical concept in the field of database structuring, it is necessary to construe the claim. I recommend adopting defendant's construction.

C. No Rewriting of Code Function

The function at issue here is found in claims 1, 21, and 32 of the '278 patent. The relevant language is as follows:

> automatically storing the applicant information entered into the second form data fields into the database by adding new records to the database, the automatic storing of the applicant information not altering the database field structure, *thereby allowing new form data fields corresponding to applicant information not previously requested to be added to an application form without requiring alterations of existing application forms or of programs that access the database,* whereby cus-

> tomized applications to different institutions share data through common, extensible data storage.

23:5-14 (claim 1) (emphasis added); *see also* 25:20-23 (similar language in claim 21); 26:25-33 (similar language in claim 32).

Also relevant to this function is language from dependent claim 2 of the '278 patent:

> The method of claim 1 in which creating a first application form customized in accordance with the preferences of the first institution includes generating a first application in accordance with stored application description information and in which a modified first application can be generated by modifying the application description information *without rewriting the computer program that creates the first application.*

23:15-22 (emphasis added). Additionally, language from dependent claim 34 is relevant:

> The method of claim 32 in which providing a database for storing information includes providing a database that is extensible *without reprogramming the program for generating the customized application,* thereby allowing an institution to readily request and store new information previously stored.

26:38-43 (emphasis added).

Defendant proposes constructions for the following terms appearing in these claims: (1) "alterations"; (2) "without requiring alterations of existing application forms or programs that access the database"; (3) "without rewriting the computer program that creates the first application"; and (4) "without reprogramming the program for generating the customized application[.]"

1. "Alterations"

Defendant proposes that "alterations" be construed as "making different in some particular, as in size, style, course, or the like; modification." Defendant bases this construction on the definition of "alter" from Webster's Encyclopedic Unabridged Dictionary of the English Language 43 (1994). Plaintiff argues that the term "alterations" is used in its plain, ordinary meaning and needs no construction. I agree with plaintiff.

2. "Without Requiring Alterations of Existing Application Forms or of Programs that Access the Database"

***32** Defendant seeks separate constructions for "without requiring alterations of existing applications" and "without requiring alterations of ... programs that access the database." As to the former, defendant proposes the following construction: "no programs for creating forms have to be rewritten, revised or reprogrammed and no forms have to be recreated or regenerated from rewritten or revised programs in order to add new form data fields to a form." For the latter, defendant proposes the following construction: "no program such as the forms engine that sends and retrieves user data to and from a database needs to be rewritten or reprogrammed in order to add new form data fields to a form."

I agree with defendant that the prosecution history shows that the patent applicants distinguished their invention from the prior art by describing a system with a "flexible forms engine that can be readily extended to handle new data fields without reprogramming the database or recreating existing forms." Exh. A to Deft's Op. Cl. Constr. Brief. Based on the distinction, the '278 patent disclosed an invention in which existing forms do not have to be altered because the forms themselves are not "hard-coded" programs that have to be rewritten. *See* 1:30-34 (noting that in prior incarnations of internet application forms, "if the institution wishes to change the application form, the institution must typically revise the source code that creates the application form, thereby making changes to the application form expensive and inconvenient.").

The problem with defendant's proposals, however, is that I see no reason not to apply my previous construction of the phrase at issue. In the *ApplyYourself* case, I construed the following phrase: "thereby allowing new form data fields corresponding to applicant information not previously requested to be added to an application form without requiring alterations of existing application forms or of programs that access the database[.]" The construction I gave the phrase was: "[n]ew form data fields corresponding to application information not previously requested could be added to an application form without altering existing application forms or programs that access the database." I also separately construed the limited phrace "programs that access the database," as "the computer software programs that retrieve user data from the database and send user data to the database." Dec. 19, 2002 Op. at pp. 34-35.

The previous constructions make clear that existing application forms and programs that access the database are not altered when new form data fields are added to the application form. This construction addresses the distinction made over the prior art in the prosecution history. Furthermore, defendant's proposed construction is cumbersome and unnecessarily wordy. The claim language itself is fairly straightforward and the prior constructions adequately define the claim limitations.

3. "Without Rewriting the Computer Program That Creates the First Application"

***33** This claim phrase is taken from dependent claim 2 of the '278 patent, quoted above. Defendant proposes the following construction: "tl.e code for the forms engine program does not have to be rewritten or reprogrammed because one has only to change the application description information that the forms engine uses to generate the application form."

Defendant cites to the specification of the '278 patent in support of its argument that "the computer program that creates the first application" is the forms engine program described elsewhere. The relevant excerpt is:

> [t]he applicant database can be extended to include new attributes without making any changes to the forms engine program or to the application files of institutions that chose not to include the new data. The forms engine automatically uses the application data file to produce the requested application in HTML format for display on the applicant's browser. The application description file can be easily modified, for example, to change labels or to add additional fields. The appearance of the application for each institution can be changed by changing its application description file, without reprogramming the forms engine.

8:60-67--9:1-3.

Plaintiff contends that the phrase "without rewriting the computer program that creates the first application" need not be construed because it is not imbued with any scientific or technical meaning and all of the words in the phrase are used in their plain, ordinary, everyday sense. Plaintiff also disputes that aspect of defendant's construction that essentially equates the "computer program" with the "forms engine program."

Plaintiff contends that defendant inappropriately imports language from the specification into the proposed claim construction to create a limitation not seen in the claim language itself. *Liebel-Flarsheim Co. v. Medrad, Inc.,* 358 F.3d 898, 904 (Fed.Cir.2004) (noting impropriety of reading a limitation from the specification into the claims). Reference to the specification is not an improper claim construction tool, because it is permissible to read the claims in light of the specification. *Id.* Thus, to the extent the specification is used as a way to confirm the apparent meaning of the claim language, the use of the specification is acceptable. Here, the use of the specification only confirms the claim language's obvious meaning that "computer program" means the forms engine program.

I start with the language of claim 2 of the '278 patent, quoted in its entirety:

> The method of claim 1 in which creating a first application form customized in accordance with the preferences of the first institution includes generating a first application in accordance with stored application description information and in which a modified first application can be generated by modifying the application description in-

formation without rewriting the computer program that creates the first application.

***34** 23:16-23. The "computer program that creates the first application" clearly refers to the first part of this claim which describes the creation of the first application form which has been customized in accordance with the preferences of the first institution. This in turn refers to a method expressed in claim 1.

In claim 1, the language provides that the method allows for the creation of, in response to a request from an applicant for an application to a first institution, a first application form customized in accordance with the preferences of the first institution. 22:37-40.

Because claim 1 discloses no further information regarding what part of the system actually creates the application to a first institution, it is necessary to examine claim 21 which describes the system used for creating and processing the forms previously disclosed in independent claim 1 and subsequent dependent claims. Claim 21 discloses a system which relies on a "forms engine program" to generate a form from the form description information. Read together, claims 21, 1, and 2 provide for the creation, by a forms engine program, of a first application form customized in accordance with the preferences of the first institution. Thus, the claim language itself supports equating the meaning of "computer program" with "forms engine program."

Accordingly, while I conclude that the remaining words in this phrase do not need construction because the words "without," "rewriting," and "the first application" are used only in their non-technical ordinary sense, I agree with defendant that "computer program" should be construed to mean forms engine program. I recognize that recommending a construction that incorporates another construed term will require the cross-referencing which I described above as unnecessarily complicating a construction. However, in this instance, the construction which relies on "forms engine program" is required because unlike the other constructions discussed above, the jury could easily apply the wrong meaning to the disputed phrase in this instance. Therefore, I recommend that the phrase "without rewriting the computer program that creates the first application" be construed as "without rewriting the forms engine program that creates the first application."

4. "Without Reprogramming the Program for Generating the Customized Application"

This claim phrase is taken from dependent claim 34 of the '278 patent, quoted above. Defendant proposes the same construction for this phrase as for the previous phrase: "the code for the forms engine program does not have to be rewritten or reprogrammed because one has only to change the application description information that the forms engine uses to generate the application form."

As with the previous phrase, defendant contends that the "program for generating the customized application" must refer to the forms engine program. In addition to the plain meaning of the claim language, defendant notes that the specification of the '278 patent indicates that the database for storing user information can be extended to include new user attributes that do not correspond to ones already in the database and that this extension does not require reprogramming the forms engine program. 7:29-35; 8:60-67-9:1-3.

***35** Claim 34 refers to the method of claim 32 which in turn, discloses a method which includes generating a customized application. Claim 34, however, does not itself disclose what actually generates that customized form. Again, it is necessary to examine claim 21 for that information. As discussed above, claim 21 discloses that the function is performed by the forms engine program. Thus, I agree with defendant that the reference to "the program for generating the customized application" in claim 34 refers to the forms engine program. This is confirmed by the specification as indicated in the previous paragraph.

I recommend that the phrase "without reprogramming the program for generating the customized application" be construed to mean "without reprogramming the forms engine program for generating the customized application."

D. Forms Processing Function

This function is expressed by the following language in claim 1 of the '042 patent:

> processing by the third party forms servicer the user information in accordance with the preferences of the institution of higher education to which the form is directed to make the user information available to the institution in a format specified by the institution, the third party forms servicer thereby providing to public users customized forms identified with institution [s] of higher education and providing to the institutions custom-formatted data, while relieving the institution of the administrative burden of processing forms and payments.

35:34-44; *see also* 36:49-57 (nearly identical language in claim 16); 38:1-9 (nearly identical language in claim 32).

In addition, in the preamble to claim 1, the patent states:

> A method of processing over a computer network forms directed by multiple public forms users to multiple institutions of higher education, the forms being processed by a third party forms servicer that is neither one of the institutions of higher education nor one of the public forms users, ...

35:2-6. This language is more or less repeated in the preambles to the other independent claims of the '042 patent. 36:32-36 (claim 16); 37:46- 50 (claim 32); 38:38-42 (claim 38).

Defendant proposes constructions for the following terms: (1) processing; (2) providing; (3) by the third party forms servicer; and (4) in a format specified by the institution.

One of the most contentious issues in the *ApplyYourself* case was the construction of the first part of the language quoted above from claim 1:

> processing by a third party forms servicer the user information in accordance with the preferences of the institution of higher education to which the form is directed to make the information available to the institution in a format specified by the institution.

I first construed this in the July 2003 summary judgment opinion, then in a subsequent August 20, 2003 Order on plaintiff's motion for reconsideration, and then again during trial in a September 3, 2003 Opinion. I finally instructed the jury that the entire phrase was to be construed as follows:

> ***36** User information provided to the institution by the servicer is available in an unlimited number of formats and is processed wholly by the third party forms servicer and not the institution. That is, the function is one of providing limitless formats for the transfer of user information from the servicer to the institution with no additional formatting or mapping performed by the institution.
>
> This construction does not preclude formatting, mapping, or other manipulation of the user information data by the institution once it is received by the institution in a format the institution specified.
>
> Any reference to "unlimited number of formats" and "limitless formats" should be interpreted to mean that the third party forms servicer provides the user information to the institution in any format specified by the institution.
>
> "in a format specified by the institution" means in any file format, and it may include any other type of format, specified by the institution.

Final Jury Instructions at p. 14 (dkt # 323). This claim construction is one of several issues from the *ApplyYourself* case currently on appeal before the Federal Circuit. Keeping this prior construction in mind, I turn to defendant's proposals.

1. "Processing"

Defendant first contends that references to "processing" in the preambles to the independent claims to "processing over a computer network forms directed by ..." and "the forms being processed by a third party forms servicer" should be construed to mean processing of the user information captured by a form rather than the form itself. I agree that the plain language of the claims supports this interpretation.

Defendant next argues that as to processing user information, the term "process" should mean "to subject to a special process or treatment (as in the course of manufacture)." Then, defendant continues, because the claim language refers to processing the user information to make it available in a format specified by the institution, the construction of "processing" must include a reference to making the information available in a format specified by the institution.

Thus, defendant argues that as it relates to *forms*, "processing" includes the step of "subjecting the user information to a special process or treatment so as to make it available to an institution in a format specified by the institution ."

I reject this proposal. Defendant's proposed construction invites confusion by referring to "special process or treatment" because such terms would themselves likely require additional construction. Furthermore, I do not agree that the verb "processing" must be construed by incorporating the phrase "making it available to an institution in a format specified by the institution." That limitation is obvious from the claim language itself. While that may be the end result of the action of "processing," it is not required as part of the construction of "processing."

***37** I have already construed the term "processing" in the context of the electronic payment function. I see no need to adopt a different construction here in the context of processing user information. Nothing in the claim language or specification indicates that the term carries different meanings in the two separate functions. Moreover, as noted above, ordinarily the same term in a claim is to be interpreted consistently. *Omega Engineering,* 334 F.3d at 1334. Thus, I propose that "process" or "processing" in the context of forms processing, be construed as "the manipulation of data within a computer system."

2. "Providing"

Defendant suggests that the term "providing" in independent claims 16, 32, and 38 of the '042 patent implies the "processing" function expressly claimed in claim 1 because these other independent claims call for the user information to be provided by the third party forms servicer to the institution in a format specified by the institution. Defendant states that the ordinary definition of provide, from Merriam Webster's Collegiate Dictionary 940 (10th ed.1994), means "to supply or make available." Defendant contends that this definition is consistent with the language of claim 1, stating that the processing step, which defendant argues is implied by "providing" in the other independent claims, is intended to "make the user information available to the institution in a format specified by the institution." Thus, defendant proposes that as it relates to *user information,* "providing" means "to make available."

Plaintiff contends that because "provide" is used only in its everyday, ordinary sense with no technical meaning having been ascribed to it, no construction is required. I agree with plaintiff.

3. "By the Third Party Forms Servicer"

The claims require the "processing" to be performed by the "third party forms servicer." The preamble for each claim recites that the "third party forms servicer" is "neither one of the multiple institutions nor one of the public form users[.]" 35:5-6; 36:36-37; 37:49-50; 38:41-42.

Defendant argues that I must construe "third party forms servicer" consistently throughout each claim because "the same word appearing in the same claim should be interpreted consistently." *Digital Biometrics,* 149 F.3d at 1345. Accordingly, defendant proposes that I adopt the same construction for "third party forms servicer," whether it be in the context of processing forms or payments. In defendant's opinion, as discussed above, the third party forms servicer which processes electronic payments is limited to the business entity hosting the forms engine software. Thus, according to defendant, the third party forms servicer which processes forms must also be limited to that same business entity hosting the forms engine software.

I do not dispute defendant's premise that the same word appearing in the same claim should be interpreted consistently. I believe that I have done that by referring to the "third party forms servicer" as the business entity hosting the forms engine software in both the electronic payment context and in the forms processing context. My construction of "third party forms servicer" as that entity remains constant throughout the claim.

***38** What is different, however, is that the function of electronic payment processing inherently requires the participation of an outside entity in the process. As explained above, at a minimum, financial institutions play a role in the processing function. Thus, while "third party forms servicer" means the business entity hosting the forms engine software, the function of processing of electronic payment information requires the participation of the third party forms servicer while contemplating the participation of an additional party.

The forms processing function does not present the same issue. There is no reason why the third party forms servicer cannot be the exclusive processor of user information. Accordingly, the third party forms servicer, when it comes to processing user information, is the sole entity involved in the process.

Thus, I recommend that the term "third party forms servicer" be construed to mean "the business entity hosting the forms engine software" no matter which function (electronic payment or forms processing) is being considered. However, I further recommend, as discussed above, that the processing of electronic payments not be limited to that entity while the processing of user information should be limited to that entity. Additionally, as before, the institution and the user/applicant are not involved in either function.

4. "Format" and "In a Format Specified by the Institution"

Given plaintiff's appeal of the prior claim construction of these phrases in the *ApplyYourself* case, plaintiff requests I defer claim construction of these phrases in this case pending the resolution of that appeal by the Federal Circuit. Acceding to plaintiff's request could unnecessarily prolong the length of this case. The case schedule for this case has, hopefully, allowed time for a decision from the Federal Circuit before trial. There is no need to defer consideration of the construction here.

Defendant states that it does not take issue with my prior construction. But, in the briefing, defendant appears to suggest that I add additional language. Defendant states that the forms processing steps, including the final step of processing the user information to make it available to an institution in a format specified by the institution, "must be construed as leaving nothing that the institution would have to do by way of processing before it can make use of the user information."

I reject the proposed construction because I think it adds nothing to the previous construction and simply uses different words to express the same meaning. When the claim limitation as I have construed it, is met, there is, by definition, nothing that the institution must do by way of processing before it can make use of the user information. That is the whole point of providing it in any format requested by the institution. The prior construction gives a more complete explanation of the concepts expressed by the claim limitation, including the concept of the institution not having to do anything related to processing, while allowing for the institution to choose to retain parts of the processing function if it desires to do so.

E. No Administrative Burden Function

***39** Independent claims 1, 16, 32, and 38 of the '042 patent refer to "relieving the institution of the administrative burden of processing forms and payments." 35:42-44; 36:55-57; 38:7-9; 38:60-61. Defendant proposes constructions for: (1) "relieving"; (2) "administrative burden"; and (3) the entire phrase "while relieving the institution of the administrative burden of processing forms and payments."

1. "Relieving"

Defendant argues that the addition of the "relieving" clause, read in the context of the claims limitations, creates a limitation on the nature and extent of the processing that a third party forms servicer must do. Defendant contends that the third party forms servicer must provide processing sufficient to *eliminate* the need for the institution to engage in processing. Defendant accurately notes that I have previously recognized that the clause creates a limiting effect in the prior construction of the phrase "in a format specified by the institution." Sept. 3, 2003 Opinion at p. 9.

Citing Merriam Webster's Collegiate Dictionary 988 (10th ed 1994) (def.1(a)), defendant argues that "relieving" means "to free from a burden." Based on this, and on defendant's contention that this ordinary meaning is consistent with the patent specification, defendant proposes to construe "relieving" as "freeing from a burden."

Plaintiff objects to the implication that "freeing" is synonymous with eliminating. Plaintiff cites the American Heritage Dictionary for the proposition that the ordinary meaning of "relieving" is to "lessen," not eliminate. Pltf's Resp. Brief at p. 30 (citing *American Heritage Dictionary* 1474 (4th ed.2000) (defining "relieving" as: "To cause a lessening or alleviation of ."). Plaintiff argues that while relieving can include eliminating or "freeing from a burden," relieving should not be limited to that meaning. Plaintiff cites to *Texas Digital,* 308 F.3d at 1202, for the proposition that "if more than one dictionary definition is consistent with the use of the words in the intrinsic record, the claim terms may be construed to encompass all such consistent meanings."

I conclude that defendant's proposal is more consistent with the interpretation of "in a format requested by the institution" than plaintiff's proposal. I start with the idea, as expressed in the September 2, 2003 Opinion in the *ApplyYourself* case, that the "thereby" clause containing the phrase "relieving the institution of the administrative burden of processing forms and payments," "acts as a summary of the function of the claim [limitation] and indicates that by providing the processed user information to the institution as custom-formatted data in a format specified by the institution, the claim will relieve the institution's burden of processing forms." Sept. 3, 2003 Op. at p. 9. I noted that the "thereby" clause was "critical to my construction" of the "in a format requested by the institution" claim phrase because "it is the relief of the burden of the institution that instruct[ed] my reading of the term 'format." ' *Id.*

*40 "Relieving," then, should properly be understood to mean the elimination of anything the institution *must* do to use the data. If there are limitations on the abilities of the third party forms servicer to provide limitless file formats and thus, a limit on its ability to provide the user information in a format specified by the institution, then the burden of processing the user information is not eliminated and thus, not relieved, because the institution then *must* do some processing to make use of the data.

While the institution may choose to do any level of "processing" whether electronic or physical, to the data received from the third party forms servicer, it cannot be *required* to do so by the inabilities of the third party forms servicer to provide the data in the format requested by the institution. The "relieving" claim term and the "in a format specified by the institution" claim phrase, are, in effect, two sides of the same coin. "Relieving," when read in the context of the claim construction for the entire "processing of user information" claim limitation, including the phrase "in a format specified by the institution," means eliminating.

This interpretation does not negate the part of the prior construction of the "processing of user information" limitation which provides that the construction "does not preclude formatting, mapping, or other manipulation of the user information data by the institution once it is received by the institution in a format the institution specified." The construction assumes that the burden on the institution of processing user information is eliminated once it receives the user information in a format it specified. The fact that the institution may choose to do additional formatting, mapping, or manipulation after that point does not suggest that the burden is not eliminated, or that by construing "relieving" as eliminating is inconsistent with this interpretation.

2. "Administrative Burden"

Defendant states that the term "administrative burden" requires no formal definition, but then it proposes the following construction: "the administrative tasks typically associated with the processing of forms such as admissions applications and any payments associated with the forms." I recommend that this term not be construed as it is used in its ordinary, customary fashion with no technical or scientific meaning revealed by the claims or the specification.

3. "While Relieving the Institution of the Administrative Burden of Processing Forms and Payments"

Based on its constructions for "relieving" and "administrative burden," defendant proposes the following construction for the entire phrase at issue:

> the institution is freed from the administrative burden of processing the relevant forms and associated payments. With respect to the processing of forms, the institution must be able to receive the user or applicant information in whatever file and other format it has specified, such that no further formatting or mapping has to be done to the data. With respect to the processing of payments, the institution must be able to receive an electronic payment credited to its account and matched to the form with which the payment is associated so that the institution is freed from the administrative burden of handling any aspect of the payment process, from verification of credit card numbers to settlement to reconciliation.

*41 Deft's Op. Brief at pp. A-3--A-4.

I recommend that this construction not be adopted. I conclude that given that the operative terms in the claim phrase have already been construed or need no construction, any additional construction is unnecessary.

G. Miscellaneous Terms

Defendant proposes constructions for two terms it refers to as "miscellaneous terms": (1) metadata; and (2) relational database.

1. "Metadata"

The term metadata appears in claims 19, 20, 36, and 37 of the '278 patent:

> 19. The method of claim 18 in which the metadata includes validation rules for the data.
> 20. The method of claim 18 in which the metadata specifies the sharing between applications or the accessibility of the data.
> 36. The method of claim 32 in which the database stores metadata describing the data.
> 37. The method of claim 36 in which the metadata describes permissible values for the data and further comprising comparing the applicant data in the completed form data fields with the permissible values.

24:47-51; 26-47-52.

Defendant cites to the following definition of "metadata" from the specification of the '278 patent: "Metadata, that is, information that characterizes the applicant data is also stored." 2:27-28. Defendant argues that metadata may describe various characteristics of the user attributes that are being stored in the database. Defendant contends that these characteristics include the properties of the fields and the relation in which the user data are arranged. Based on these arguments, defendant offers the following construction for "metadata": "information that describes user data including the properties of the fields and the relation in which user data are arranged."

Plaintiff takes issue with defendant's proposal. In contrast to defendant's proposal, plaintiff offers the following: "information that describes data." As plaintiff notes, the parties agree that metadata is information that describes data.

Plaintiff opposes defendant's inclusion of "user data." Plaintiff states that the limitation of "user data" is not in the claim. Plaintiff indicates that the specification makes clear that metadata can be used to describe parameters, such as validation criteria for data, rather than describing user data. Plaintiff cites to the following part of the specification:

> Metadata, that is, information that characterizes the applicant data is also stored. For example, in one embodiment, an attribute table describes characteristics, such as permissible values and accessibility to various institution personnel, of applicant attribute data. In another embodiment, such properties of the applicant attribues are stored in XML files. Storing metadata provides greater control over the data validation, sharing between forms, grouping, and access.

2:30-37. Based on this reference, plaintiff argues that defendant's restriction to user data is inconsistent with the ordinary meaning and is contrary to the intrinsic evidence. Furthermore, plaintiff contends that defendant's proposal that the "user data" must also include "properties of the field" and "the relation in which user data are arranged," is not required.

***42** I agree with plaintiff. As to "user data," even claim 19 itself seems to contradict defendant's proposal by stating that the "metadata includes validation rules for the data." Additionally, the specification reference cited by plaintiff suggests that while metadata indeed includes information characterizing the applicant data, the "characteristics" include "permissible values" and "accessibility to various institution personnel." These encompass more than "user data." Thus, "metadata" it is not restricted to "user data."

Another problem is defendant's references to "field" and "relation" which suggest that "metadata" is information about data as it exists in a two-dimensional table. For the reasons described above, this is inconsistent with the patent's specification and basic claim construction standards which caution against limiting a claim term to its preferred embodiment. Accordingly, I recommend that plaintiff's proposal for "metadata" be adopted.

2. "Relational Database"

The term appears in claims 17, 31, and 39 of the '278 patent:

> 17. The method of claim 1 in which the database includes a relational database or XML data.
>
> 31. The system of claim 21 in which the first or second data storage comprises one or more relational database tables stored on a computer readable medium.
>
> 39. The method of claim 32 in which the database includes a relational database.

24:43-44; 25:58-60; 26:56-57.

Defendant proposes the following construction for the term: "a database organization method that links files together as required." Plaintiff proposes the following construction: "a database that is organized in a manner than can link tables or records together as required."

Obviously, the two proposals are similar. Plaintiff argues that its alternative uses the terminology "tables or records" instead of "files" because that is grammatically consistent with the claim language and is more accepted in the industry. I agree with plaintiff and recommend that plaintiff's proposal be adopted.

H. Performance of Method Claims in Order Recited

Defendant argues that the method claims of both patents (independent claims 1 and 32 of the '278 patent and independent claims 1, 16, and 38 of the '042 patent), should be construed to require that the steps set forth be performed in the order in which they are recited. Plaintiff asserts that it would be technologically possible to achieve the purpose of the claims even if many of the claim steps were performed in a sequence different than that recited in the claims.

"Unless the steps of a method actually recite an order, the steps are not ordinarily construed to require one." *Interactive Gift Express, Inc. v. Compuserve Inc.*, 256 F.3d 1323, 1342 (Fed.Cir.2001). However, requiring the performance of the steps of a method in the order recited may "ensue when the method steps implicitly require that they be performed in the written order." *Id.*

***43** A two-part test is used for "determining if the steps of a method claim that do not otherwise recite an order, must nonetheless be performed in the order in which they are written." *Altiris, Inc. v. Symantec Corp.*, 318 F.3d 1363, 1369 (Fed.Cir.2003). First, the court looks to the "claim language to determine if, as a matter of logic or grammar, they must be performed in the order written." *Id.* "If not, we next look to the rest of the specification to determine whether *it* directly or implicitly requires such a narrow construction." *Id.* at 1370 (internal quotation omitted). "If not, the sequence in which such steps are written is not a requirement." *Id.*

1. Claim 1 of the '278 Patent

This claim discloses the following steps, listed in the order they are recited in the claim:

(1) creating a first application form to a first institution in response to a request from an applicant, the form customized to the preferences of the first institution and including first form data fields for entering applicant information;

(2) presenting this first application form to the applicant over a computer network;

(3) entering applicant information in the first form data fields;

(4) posting the applicant information entered in the first form data fields to a server;

(5) storing the posted applicant information in a database (with more info about the database disclosed);

(6) creating a second application form to a second institution in response to a request from "the" applicant (as opposed to "a" applicant disclosed in the first step addressing the first application form), the form customized to the preferences of the first institution and including second form data fields for entering applicant information, at least one which corresponds to applicant information not entered into the first form data fields;

(7) automatically inserting into some of the second form data fields applicant information from the database;

(8) providing the second application form to the applicant over a computer network;

(9) entering applicant information into the second form data fields into which information was not inserted from the data storage or into which the data inserted from the data storage is to be changed;

(10) posting the applicant information entered into the second form data fields to the server; and

(11) automatically storing the applicant information entered into the second form data fields into the database.

I conclude that as a matter of logic, the steps of this claim implicitly require that they be performed in the written order. The invention cannot provide the first application to the applicant in step (2) without first creating the first application in step (1). The applicant cannot enter the applicant information in the first application form data fields in step (3), without having been provided the first application in step (2). The applicant information in the first form data fields cannot be posted to the server in step (4), unless it has been entered in step (3). The applicant information cannot be stored in the database in step (5), unless it has been posted in step (4).

***44** A second application form disclosed in step (6) cannot be a "second" application form unless there has already been a "first" application form recited in steps (1)--(5). Additionally, the reference in step (6) is to "the" applicant, with the antecedent basis clearly being the applicant who has filled out the first application form data fields in steps (1)--(5). For the server to recognize "the" applicant, "the" applicant's information from the first form must have been stored in the database in step (5).

The next step (7) discusses the automatic populating of data stored in the database as a result of steps (1)--(5), into the data fields in the second application created in step (6). Thus, because the second application needs to be created in step (6) before this automatic populating of data into that second application form, and because the first application data has to have been stored in the database as a result of steps (1) through (5), then logically, steps (1) through (5) must occur before step (6) and (6) must occur before (7).

Step (7) must occur before step (8). The automatic population of the second form data fields must occur before the second form is provided to the applicant because if the second form data fields were not automatically populated before the applicant received the second form, one of the main purposes of the invention would be defeated, and because if the second form is provided to the applicant before the automatic population of the second form data fields, there appears to be no step to trigger the automatic insertion of this data. Accordingly, steps (1) through (8) must occur in sequence.

Next, step (8), which provides the second application form, must occur before step (9) which is where the applicant enters applicant information onto the second form. Then, step (9), entering the new applicant information into the second form, must occur before the posting of the entered applicant information to the server in step (10), and the posting in step (10) must occur before the information is stored in the database in step (11).

Thus, as for claim 1 of the '278 patent, I recommend concluding that the steps must be performed in the order recited.

2. Claim 32 of the '278 Patent

This claim discloses the following steps, listed in the order they are recited in the claim:

(1) providing at least two application infoımation files, each describing a customized application for an institution;

(2) providing a database for storing applicant information entered on an application and for providing applicant information for inserting into subsequent applications;

(3) generating a customized application in response to a request over a computer network from an applicant, the application form and content being specified by one of the at least two application information files, the application including multiple form data fields for entering applicant information;

(4) populating the form data fields of the customized data fields of the customized application using applicant information from the database;

***45** (5) transmitting the customized application over a computer network to a requesting applicant;

(6) completing form data fields of the application that were not populated with applicant information from the database; and

(7) automatically storing the applicant information from the database.

Here, I disagree with defendant that all of the steps in this claim must be performed in the order recited. I conclude that this method claim could start with either step (1), providing at least two application information files, or step (2), providing a database for storing applicant information. The claim language and specification suggest no reason why either one of these steps must precede the other.

However, the claim language makes clear that step (1), providing at least two application information files, must occur before step (3) which requires the generation of a customized application based on the specifics in one of the at least two application information files.

The claim language also makes clear that step (4), regarding populating the form data fields of the customized application with applicant information in the database, must occur after both steps (2) and (3). Step (4) requires the information contained in the database outlined in step (2) and it also requires the customized application recited in step (3).

For the reasons explained above in the context of steps (7) and (8) of claim 1 of the '278 patent, step (4), addressing the population of the data fields using applicant information in the database, must precede step (5) in which the customized application is transmitted to a requesting applicant.

Steps (1) through (5) must precede step (6) because step 6 requires the applicant to enter information into the data fields of the customized application that were not automatically populated with information stored in the database. And, step (7) must be preceded by step (6) because it requires storing the information that was entered in step (6).

Thus, I recommend concluding that while steps (1) and (2) must precede step (3) and steps (3) through (7) must occur in the order recited, step (1) and step (2) could be performed with either one preceding the other.

3. Claim 1 of the '042 Patent

This claim discloses the following steps, listed in the order they are recited in the claim:

(1) presenting to a form user over a computer network by a third party forms servicer in response to a request by the form user, a form directed to one of multiple institutions of higher education, the form being generated by a forms generator that generates multiple customized forms;

(2) entering user information onto the form;

(3) entering payment information;

(4) the third party forms servicer receiving user information and electronic payment information entered by the user;

(5) processing of an electronic payment by the third party forms servicer; and

(6) processing the user information by the third party forms servicer.

***46** I recommend concluding that the claim language of this claim logically requires the steps to be performed in the sequence in which they are recited except that step (3) could be performed before or after step (2), and steps (5) and (6) could precede each other as long as they follow step (4).

4. Claim 16 of the '042 Patent

This claim discloses the following steps, listed in the order they are recited in the claim:

(1) presenting to a form user over a computer network by a third party forms servicer a form directed to one of the multiple institutions, the forms including fields for the forms user to enter information;

(2) receiving by the third party forms servicer over the computer network user information and electronic payment information entered by the user;

(3) processing by the third party forms servicer an electronic payment associated with the form;

(4) providing by the third party forms servicer the user information to the institution to which the form is directed in a format specified by the institution.

Here, step (1) has to precede step (2) because the presentation of the form to the user has to occur before the third party forms servicer can receive any user information or electronic payment information entered by the user. Because step (2) contains the receipt of both the user information and the electronic payment information, there is no way to separate those functions in this claim and it seems clear that the receipt of the information in step (2) must occur before the third party forms servicer can either process the electronic payment information as stated in step (3) or before it can provide the user information to the institution in step (4).

This, I recommend concluding that the steps in claim 16 must be performed in the order in which they are recited.

5. Claim 28 of the '042 Patent

This claim discloses the following steps, listed in the order they are recited in the claim:

(1) receiving by an institution from a third party forms servicer user information in format ..., the user information being derived from a form customized for the institution, ...;

(2) receiving from the form user via the third party form servicer an electronic payment associated with the customized form; and

(3) thereby providing to the form user a customized form identified with the institution and providing the institution with custom formatted data and electronic payment.

Clearly, steps (1) and (2) have to occur before step (3). However, step (1) does not necessarily need to precede step (2) as they both involve the institution receiving either user information or an electronic payment from or via the third party forms servicer. These could happen simultaneously for example.

While step (1) and step (2) must precede step (3), because (1) and (2) could occur simultaneously, I recommend concluding that the steps in claim 38 do not need to be performed in the order in which they are recited.

CONCLUSION

***47** I recommend construing the claims as discussed in this Findings & Recommendations and concluding that the steps in claim 1 of the '278 patent and the steps in claim 16 of the '042 patent must be performed in the order recited.

SCHEDULING ORDER

The above Findings and Recommendation will be referred to a United States District Judge for review. Objections, if any, are due November 19, 2004. If no objections are filed, review of the Findings and Recommendation will go under advisement on that date.

If objections are filed, a response to the objections is due December 3, 2004, and the review of the Findings and Recommendation will go under advisement on that date.

IT IS SO ORDERED.

Not Reported in F.Supp.2d, 2004 WL 2429843 (D.Or.)

END OF DOCUMENT

H
Only the Westlaw citation is currently available.

United States District Court, D. Delaware.
PIPE LINERS, INC. and Hydro Conduit Corporation, Plaintiffs,
v.
PIPELINING PRODUCTS, INC., Defendant.
No. Civ.A. 98-164(SLR).

Oct. 22, 1999.

Richard D. Kirk, of Morris, James, Hitchens & Williams, Wilmington, Delaware, Bradley B. Geist, Louis S. Sorell, David T. Cunningham, Richard L. Blaylock, and Gary Butter, of Baker & Botts, LLP, New York, New York, for Plaintiffs, of counsel.

Douglas E. Whitney, of Morris, Nichols, Arsht & Tunnell, Wilmington, Delaware, Edward V. Filardi, Benjamin S. Lee, and Vincent Filardo, Jr., of White & Case, L.L.P., New York, New York, for Defendant, of counsel.

MEMORANDUM OPINION
ROBINSON, J.

I. INTRODUCTION

***1** Plaintiffs Pipe Liners, Inc. and Hydro Conduit Corporation filed this suit against defendant Pipelining Products, Inc. on April 2, 1998 alleging infringement by defendant of U.S. Patent No. 4,985,196 ("the '196 patent"), which discloses a method for *in situ* installation of thermoplastic liner within a host pipe. Before the court is defendant's motion for summary judgment. (D.I.15) For the following reasons, the court shall grant in part and deny in part defendant's motion.

II. BACKGROUND

The '196 patent, of which plaintiff Pipe Liners is an assignee, discloses a method for lining pipes with a thermoplastic tube. The patent principally addresses the need for an economical and efficient means to fix damaged underground pipes, specifically sewer pipes. As underground sewer pipes age, they tend to crack and leak sewage into the ground. (D.I. 17 at A382) Excavating and replacing these pipes is often expensive and causes disruption and inconvenience. Prior to the issuance of the patent in suit, a French inventor named Jacques Laurent developed and patented a method of repairing damaged pipes without having to excavate them. Because the Laurent patent discloses technology similar to that at issue in this case, a brief discussion of the Laurent patent will facilitate analysis of the '196 patent.

The Laurent patent, which is prior art in the present suit, utilizes the "elastic memory" properties of thermoplastics to line underground pipe. This elastic memory property allows manufacturers to heat a thermoplastic conduit to a given temperature, called the shape memory activation temperature, and deform the plastic into a different shape. Although the thermoplastic tends to remain in this deformed state, it keeps a "memory" of its original undeformed shape. If the thermoplastic is reheated to its shape memory activation temperature, it will return to its initial dimensions. The Laurent patent discloses a method of deforming a circular thermoplastic tube at its shape memory activation temperature into a reduced, U-shaped cross section, which facilitates insertion of the tube into a length of host pipe. Thereafter, the deformed thermoplastic tube may be reheated to its shape memory activation temperature, which causes the tube to return to its initial circular dimensions and, thus, conform to the interior walls of the damaged pipe. The Laurent process thereby creates a new, plastic pipe within an existing host pipe without need of significant excavation.

The applicants for the '196 patent characterize their invention as an improvement on the Laurent process. (D.I.17, Ex. 2, col.1, lns.16-19) The ' 196 patent's specification describes the invention as providing

> an improved method and apparatus for installing temporarily deformed pipe liner within a

> pipeline, expanding the deformed liner to its original cylindrical shape, taking additional steps causing the liner to conform even more precisely to the interior contour of the pipe, and flaring opposite ends of the liner into engagement with respective radially directed pipe flanges.

***2** (D.I.17, Ex. 2, col.1, lns.57-64) (emphasis added). These "additional steps" refer to a combination of heat and pressure that causes the liner to expand further and fit snugly within the host pipe, thus eliminating the problems of "annular or other pockets of air between the liner and the inner pipe wall" associated with the Laurent pipelining process. (D.I. 17, Ex. 2, col 3, lns. 35-37)

The '196 patent contains two independent claims (claims 1 and 12) and eighteen dependent claims. The Patent and Trademark Office issued Reexamination Certificate B1 4,985,196 on November 18, 1997, which added dependent claims 21 and 22. (D.I. 17, Ex. 2, at A016) In the preferred embodiment, the pipe liner is constructed of high density polyethylene ("HDPE") and, at installation, temporarily deformed into a U-shaped cross section. [FN1] (*See, e.g.,* D.I. 17, Ex. 2, Fig. 5, at A005) The specification of the '196 patent also incorporates by reference U.S. Patent No. 4,863,365 ("the '365 patent"), which discloses "a specific method and apparatus for manufacturing temporarily deformed thermoplastic conduit." (D.I.17, Ex. 2, col.1, lns.43-45) The specification of the '196 patent explains that the invention may be practiced such that "the liner is temporarily collapsed in a manner described in [the '365 patent]." (D.I.17, Ex. 2, col.1, ln.68-col.2., ln.4) The manner of temporarily collapsing thermoplastic tubing is described by the '365 patent as follows:

> FN1. Plaintiffs' pipelining method is known by the trade name U-Liner®)). (D.I. 17, Ex. 2, col. 4, ln. 15; Ex. 3 at A382)

> It is an object of this invention to deform an initially extruded tubular cross section without adverse effect on its structural integrity, and in such a manner that its initially extruded cross section can be restored. To this end, controlled heat is applied to establish a softened condition of the thermoplastic material after its extrusion, while simultaneously applying deforming tools thereto in order to reduce its cross sectional configuration.

(D.I. 65, Tab 1, Ex. C, col. 1, lns. 52-60) The specification of the '365 patent then provides an example of a method for deforming an extruded thermoplastic liner:

> The extruder means E is state of the art and receives the raw thermoplastic material and forces it through a[n] extrusion die 17 at, for example, 250°>>>to 300° F, using heating means 18 to attain that temperature. The cooling means C1 is state of the art, and preferably a vacuum cooling means supported by a vacuum cooling unit 19 and reducing the tube form temperature to, for example, 160° F. The deformer apparatus D is subject to heating means H that maintains the desired deformation temperature of, for example, 160° F.... [I]t is to be understood that the aforementioned temperatures can vary as circumstances require.

(D.I. 65, Tab 1, Ex. C., col. 5, lns. 6-16, 20-22)

Independent claims 1 and 12 of the '196 patent teach essentially the same process of installing a thermoplastic liner in a host pipe. Initially, plaintiffs alleged infringement of both claims, but plaintiffs since have advised defendant that they will not assert claim 1 of the '196 patent. (D.I. 64 at 1-2) Thus, only claims 12-22 of the '196 patent are at issue. Unlike the '365 patent's specification, claim 12 of the '196 patent does not provide specific temperatures at which the deformation of the thermoplastic occurs. Claim 12 teaches

> ***3** a process for installing in situ a thermoplastic liner in a generally horizontally extending, generally cylindrical pipe, comprising the steps of:
> (a) providing an elongate hollow liner formed of thermoplastic material having a cross-section altered at a shape memory activation temperature from a generally cylindrical cross-section having an original outer diameter substantially comparable to the inside diameter of the pipe to be lined to a reduced cross-section having reduced cross-

sectional dimensions to enable the liner to be pulled into the pipe, said liner in said altered cross-section having a predetermined wall thickness;

(b) inserting said altered liner into said pipe such that end portions of said liner extend beyond opposite ends of said pipe;

(c) partially expanding the liner ends portions which extend beyond the opposite ends of the pipe by mechanical means inserted into said liner end portions such that said expanded liner end portions approximate the original cylindrical shape of said liner;

(d) sealing the expanded liner end portions beyond the opposite ends of said pipes to seal the interior of said liner at its opposite ends;

(e) subsequent to the step of sealing the liner and while maintaining the liner sealed, generally conforming said liner to the interior wall of the pipe while maintaining substantially the original predetermined wall thickness by (1) injecting a heated fluid into and through said sealed liner, (2) pressurizing the interior of said liner to a first predetermined pressure above atmospheric pressure by means of said heated fluid and (3) reheating said liner to a predetermined temperature by heat transfer from said heated fluid to said liner, whereby, the liner returns substantially to its remembered cylindrical cross-section; and

(f) then increasing the pressure in said liner to a second predetermined pressure above said first predetermined pressure to conform the liner substantially precisely to the interior wall surface contours of the pipe; and

(h) while the liner is still hot, introducing a cooling fluid into the liner for flow therethrough to fix the liner in final form in conformance to the interior wall of the pipe. [FN2]

(D.I.17, Ex. 2, col.10, lns.60-68, col.11, lns.1-37, col.12, lns.1-2)

In its motion for summary judgment, defendant argues that its pipelining process, known by the trade name Sure-Line®, does not infringe claim 12 of the '196 patent either literally or under the doctrine of equivalents because the Sure-Line® process does not: (1) deform the thermoplastic conduit "at a shape memory activation temperature;" (2) utilize "a second predetermined pressure" to substantially conform the liner to the host pipe wall; (3) cool the liner following a second predetermined pressurization stage; or (4) have a liner with a diameter "substantially comparable to" the inside diameter of the pipe to be lined. (D.I. 17, Ex. 2, col. 10, ln. 65; col. 11, lns. 31-32; col. 10, lns. 67-68) Defendant also contends that claim 12 is invalid on its face for omitting an essential element of the disclosed invention. Before addressing each of these arguments, the court first must construe the disputed claim language.

III. CLAIM CONSTRUCTION

***4** It is the court's "power and obligation to construe as a matter of law the meaning of language used in the patent claim." *Markman v. Westview Instruments, Inc.,* 52 F.3d 967, 979 (Fed.Cir.1995), *aff'd,* 517 U.S. 370 (1996). The principles of claim construction are well established. The exercise begins with the claim language, which defines the scope of the claim. *See York Prods., Inc. v. Central Tractor Farm & Family Ctr.,* 99 F.3d 1568, 1572 (Fed.Cir.1996). In analyzing claim language, the court must employ "normal rules of syntax," *Eastman Kodak Co. v. Goodyear Tire & Rubber Co.,* 114 F.3d 1547, 1553 (Fed.Cir.1997) for "[a] claim must be read in accordance with the precepts of English grammar." *In re Hyatt,* 708 F.2d 712, 714 (Fed.Cir.1983). The court also must ascribe to any technical term used in a claim "the meaning that it would be given by persons experienced in the field of the invention, unless it is apparent from the patent and the prosecution history that the inventor used the term with a different meaning." *Hoechst Celanese Corp. v. BP Chems., Ltd.,* 78 F.3d 1575, 1578 (Fed.Cir.1996).

In order to give context to the claim language, the court also must review the specification. The Federal Circuit has explained that

> [t]he specification acts as a dictionary when it ex-

FN2. There is no element (g) to claim 12.

> pressly defines terms used in the claims or when it defines terms by implication.... As we have repeatedly stated, "claims must be read in view of the specification, of which they are a part." ... The specification contains a written description of the invention which must be clear and complete enough to enable those of ordinary skill in the art to make and use it. Thus, the specification is always relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of the disputed term.

Vitronics Corp. v. Conceptronic, Inc., 90 F.3d 1576, 1582 (Fed.Cir.1996).

The last source of intrinsic evidence relevant to claim construction is the prosecution history of the patent where, as here, it is in evidence. The prosecution history contains the complete recoid of all the proceedings before the Patent and Trademark Office, "including any express representations made by the applicant regarding the scope of the claims." Id. at 1583. The prosecution history, therefore, "is often of critical significance in determining the meaning of the claims." *Id.* Extrinsic evidence of claim meaning, on the other hand, is improper in most instances. *See id.* Extrinsic evidence includes expert testimony.

A. "Shape Memory Activation Temperature"

1. Plaintiff's Proposed Construction

Claim 12(a) of the '196 patent requires that the elongate, hollow thermoplastic liner have a cross-section altered at "a shape memory activation temperature." (D.I.17, Ex. 2, col.10, ln.65) Neither the claims nor the specification of the '196 patent provide a specific temperature associated with the term "shape memory activation temperature." Plaintiffs argue for a functional interpretation of that term, which would include a range of temperatures--including any temperature "above ambient temperature, which still allows the deformed liner to remain sufficiently deformed for insertion of the liner into a host pipe." (D.I. 127 at 33)

*5 As an initial matter, the court notes that neither plaintiffs nor the '196 patent provide a definition of "ambient." Plaintiffs appear to use "ambient" to refer to normal, "room" temperatures, but they do not specify what those temperatures might be. Commonly understood, "ambient" means "[t]he temperature of the environment in which an experiment is conducted or in which any physical or chemical event occurs." Richard J. Lewis, Sr., *Hawley's Condensed Chemical Dictionary* 48 (13th ed.1997). Ambient, then, could refer to a wide array of temperatures and, if applied to the construction of "shape memory activation temperature," it would deprive that claim limitation of any meaning.

Nonetheless, plaintiffs point to the claim language itself for support of this functional interpretation of "shape memory activation temperature." Plaintiffs argue that the term "shape memory activation temperature" is limited "functionally, namely to enable the altered liner to be pulled into the host pipe." (D.I. 128 at 8) The claim language "to enable the liner to be pulled into the pipe," however, does not modify "shape memory activation temperature;" rather, it modifies "having reduced cross-sectional dimensions." (D.I. 17, Ex. 2, col 11, lns. 2-3) Thus, the claim language itself sheds no light on the scope of the term "shape memory activation temperature." [FN3]

Plaintiffs also rely heavily on the '365 patent's specification as support for their functional interpretation of shape memory activation temperature. Spe-

[FN3] FN3. For these reasons, the court also declines to accord any weight to the testimony of plaintiffs' expert. Plaintiffs' expert, Stanley Mruk, construed the term "shape memory activation temperature" to mean "any temperature above ambient which enables the liner to be temporarily deformed while retaining a memory of the liner's original shape (reformed) by heating the pipe liner to a temperature at or above the temperature used in the deformation of the pipe liner." (D.I. 67 at 4, ¶ 9) Mruk's construction thus rests on an undefined term (i.e., "ambient") as well as an implausible reading of the claim language.

cifically, plaintiffs point to that patent's description of "a desired deformation temperature of, for example, 160° F." (D.I. 65, Tab 1, col. 5, lns. 15-16) Plaintiffs contend that the '196 patent's incorporation of the '365 patent compels the conclusion that shape memory activation temperature "can vary as circumstances require." (D.I. 65, Tab 1, Ex. C., col. 5, lns. 21- 22) A careful reading of the '365 patent reveals, however, that the term "shape memory activation temperature" never appears therein. Indeed, both the 160° F cited by plaintiffs as a shape memory activation temperature and the phrase, "temperatures [that may] vary as circumstances require," relate merely to the "desired deformation temperature" of the preferred embodiment described in the '365 patent's specification. (D.I. 65, Tab 1, Ex. C ., col. 5, ln. 15) Other than plaintiffs' conclusory assertion, there is no basis for equating the '365 patent's use of the term "deformation temperature" with the '196 patent's specific use of "shape memory activation temperature"--a highly technical term capable of precise calculation. (*See, e.g.*, D.I. 17, Ex. 3 at A363 (providing formula for crystallization temperature, which the coinventor equates with shape memory activation temperature at D.I. 17, Ex. 3 at A351)).

Because the meaning of the term "shape memory activation temperature" is not apparent from the patent, the court must turn to the prosecution history to determine the "meaning that it would be given by persons experienced in the field of invention." *See* *Hoechst Celanese Corp., 78 F.3d at 1578*. The prosecution history reveals that plaintiffs took pains to convince the Examiner that "shape memory activation temperature" referred to a specific range of temperatures, identifiable by calculation or by reference to the "specification sheet" of particular thermoplastics, which lists the thermoplastic's crystallization point.

2. The Prosecution History

***6** Because the prosecution history plays such a significant role in determining the meaning of "shape memory activation temperature," the court shall review it in considerable detail. Plaintiffs filed their initial patent application in October of 1987. In this initial application, the proposed specification of the patent referred to 160° F as the temperature at which deformation of the thermoplastic liner occurred. (D.I. 17, Ex. 3 at A025) Claim 1 of the initial application also disclosed a process for "altering the cross-sectional shape of the liner to reduce the cross-sectional dimension thereof at a shape memory activation temperature of about 160-180° F." (D.I. 17, Ex. 3 at A044) In July of 1988, the Examiner denied claims 1 through 18 of the application as obvious in light of the Laurent patent and other prior art. (D.I. 17, Ex. 3 at A082-088)

In response, plaintiffs amended claim 1 to add, "so as to permit the liner to be pulled into the pipe," following that claim's recitation of a shape memory activation temperature "of about 160-180° F." (D.I. 17, Ex. 3 at A102) In February of 1989, however, the Examiner again rejected claim 1 for, among other reasons, obviousness. (D.I. 17, Ex. 3 at A112-118) Plaintiffs again amended their application in June of 1989 and canceled claim 1 along with other claims. Plaintiffs then added several new claims. Of these new claims, claim 31 disclosed a process for installing thermoplastic pipe and

> (b) altering the cross-sectional shape of the liner to reduce the cross-sectional dimension thereof at a shape memory activation temperature of about 221-277° F so as to permit the liner to be pulled into the pipe....

(D.I. 17, Ex. 3 at A141) Dependent claim 35 of these new amendments also disclosed "[a] process according to Claim 31 wherein said shape memory activation temperature is about 260° F." (D.I. 17, Ex. 3 at A142) Similarly, claim 36 and its dependent claim 40 each disclosed shape memory activation temperatures of "about 221-277° F" and 260° F, respectively. (D.I. 17, Ex. 3 at A143-44)

Plaintiffs also added claim 41, the predecessor of claim 12 of the '196 patent. Initially, claim 41 taught merely "an elongate hollow liner formed of thermoplastic material having a cross-section altered at a shape memory activation temperature...." Although claim 41 did not provide a specific shape memory activation temperature, its dependent claims 47 and 50 each provided, "[a] process according to Claim 41, wherein said memory activation temperature is within a range of 221-277° F." (D.I. 17, Ex. 3 at A146)

Significantly, in their remarks to the preceding amendments, plaintiffs' patent counsel explained that

> the reference in the specification and claims to 160° F as the melting temperature of the polyethylene liner material is incorrect.... Applicants enclose five specification sheets from various companies indicating that the melting temperature of polyethylene was a known parameter at a time prior to the date of this application and specifically known to lie within a range of 221-277° F.

*7 (D.I. 17, Ex. 3 at A147) It appears that plaintiffs' patent counsel equated "melting temperature" with "shape memory activation temperature." [FN4] Each of the specification sheets referred to by plaintiffs' patent counsel pinpoint the melting temperature of the various thermoplastics as above 200°> F. (D.I. 17, Ex. 3 at A163-67) Plaintiffs' patent counsel also submitted a declaration of one of the coinventors in which the coinventor confirmed that these specification sheets indicated melting points in the range of 221-277° F. (D.I. 17, Ex. 3 at A162)

> FN4. Later in the prosecution, however, the '196 patent's coinventor equates shape memory activation temperature with a thermoplastic's "crystallization temperature" and provided the Examiner with a formula for calculating that temperature. (D.I. 17, Ex. 3 at A363; *see also* A351)

Despite these amendments, the Examiner again rejected the newly added claims in September of 1989. (D.I. 17, Ex. 3 at A296-302) In rejecting the claims for, among other reasons, obviousness, the Examiner noted:

> It is submitted that the steps of altering the tube's cross-section at 210°>> F and heating to expand the tube in British Application -695 are inherently at the thermoplastic liner's shape memory activation temperature as recited in the instant claims. This temperature is dependent on particular material used.

(D.I. 17, Ex. 3 at A298) (emphasis added). Following this rejection of their newly added claims, plaintiffs again offered amendments to their application in December of 1989. (D.I. 17, Ex. 3 at A303-12) In their remarks to these amendments, plaintiffs' patent counsel distinguished the aforementioned "British Application -695" by noting that, unlike the British Application, "applicants require the cross-sectional shape of the liner to be reduced by altering such shape at a shape memory activation temperature of about 221-277° F." (D.I. 65, Ex. 3 at A310) The British Application disclosed a process of deforming polyvinyl chloride ("PVC") tubing at "approximately 210° F." (D.I. 17, Ex. 3 at A173) Plaintiffs' patent counsel argued strenuously that, "PVC does not have shape memory characteristics, and .... the temperature to which the liner is elevated in [the British Application] is not within the range claimed...." (D.I. 17, Ex. 3 at A310) Contrary to plaintiffs' current contention that shape memory activation temperature is a variable, "functional concept," plaintiffs' patent counsel also acknowledged that, "a shape memory activation temperature is a known property, for example, of polyethylene material, prior to this invention." (D.I. 17, Ex. 3 at A309)

In January of 1990, the Examiner rejected plaintiffs' application on the ground that the recitation of a shape memory activation temperature "of about 221-277° F" constituted new matter. Further, the Examiner remarked that this "range would read on any number of polymers dependent on their exact composition." (D.I. 17, Ex. 3 at A325-26) In response to the Examiner's new matter objections, plaintiffs amended their application in April and again in May of 1990. (D.I. 17, Ex. 3 at A331-39; A340-61) In these amendments, plaintiffs deleted reference to 160° F in the specification as the deformation temperature and replaced it first with 260° F and, in May, with 235° F. Plaintiffs' patent counsel explained this change in his remarks to the May 1990 amendments:

> *8 [I]t must be recognized that patent specifica-

> tions are directed to those skilled in the art. A person skilled in this art would recognize 160°>> F as an incorrect shape memory activation temperature simply by reference to the specification sheet for this particular type of material specifically disclosed as the preferred embodiment and available at the time of this filing. The person of ordinary skill in the art would be advised by such specification sheet of the actual shape memory activation temperature.
>
> Also, materials such as nylon, Teflon and ABS are disclosed ... together with the Union Carbide material. All of those materials have shape memory activation temperatures above 200° F as indicated on the additional specification sheets for each of those materials accompanying the Declaration of [the coinventor]. Consequently, a person of ordinary skill in the art would recognize that the temperature of 160° F could not be the shape memory activation temperature and would be directed by those specification sheets to the appropriate shape memory activation temperature.

(D.I. 17, Ex. 3 at A351-52)

Also in May, plaintiffs canceled claims 31-35 and claim 47 of the application and amended claims 36 and 41. Plaintiffs removed from claims 36 and 41 [FN5] any reference to a specific shape memory activation temperature. Thus, in May of 1990, claim 41 disclosed in relevant part:

> [A]n elongate hollow liner formed of thermoplastic material having a cross-section altered at a shape memory activation temperature from a generally cylindrical cross-section having an original diameter substantially comparable to the inside diameter of the pipe to be lined to a reduced cross-section having reduced cross-sectional dimensions to enable the liner to be pulled into the pipe, whereby the liner is maintained in its reduced cross-sectional shape with substantially no tendency to return to its cylindrical cross-section and retains a memory of its cylindrical cross-section, said liner in said altered cross-section having a predetermined wall thickness....

(D.I. 17, Ex. 3 at A343)

In a final supplemental amendment filed in August of 1990, plaintiffs deleted all reference to a specific shape memory activation temperature from the specification and amended claim 41 to read as claim 12 now reads in the '196 patent. (D.I. 17, Ex. 3 at A431-38) In explaining these changes, plaintiffs' patent counsel remarked:

> Applicants have attempted to amend the specification to present the proper numerical temperature but without apparent success. Thus, by canceling the numerical temperature for the shape memory activation temperature, the patent issuing from this application will not be misleading and, of course, the actual value is disclosed in the file wrapper. The actual numerical temperature is also not necessary to the claims inasmuch as those claims do not specify the precise numerical shape memory activation temperature.

(D.I. 17, Ex. 3 at A437) (emphasis added).

*9 Thus, the prosecution history reveals (1) that shape memory activation temperature is a specific temperature defined by the particular properties of the thermoplastic at issue, (2) that plaintiffs contemplated shape memory activation temperatures that were well above ambient temperature (assuming "ambient" refers to normal, "room" temperatures), and (3) that the patentee defined shape memory activation temperature first (and, apparently, erroneously) as a given thermoplastic's "melting point" and later as a thermoplastic's crystallization point. In defining shape memory activation temperature as a thermoplastic's crystallization point, the '196 patent's co-inventor declared that:

> [4. The preferred embodiment's] crystallization point is given as 113°>> C, or about 235° F. This is the memory activation temperature for that particular material.
>
> 5. The crystallization temperature is a temperature defining the maximum crystallization speed and it may be obtained by the formula T_c << con-

FN5. Claims 36 and 41 were renumbered as claims 1 and 12, respectively, upon issuance of the '196 patent. (D.I. 17, Ex. 3 at A442)

> gruent>>>>>> (T_m + T_g)/2 where T_c is the crystallization temperature, T_m is the melting temperature and T_g is the glass transition temperature. Each of nylon, Teflon and ABS, as disclosed in this application as an alternative material to the specifically identified and preferred Union Carbide material has a crystallization temperature above 200°>> F. This is evidenced by calculations and specification sheets for those materials....

(D.I. 17, Ex. 3 at A363-64) (emphasis added). The prosecution history thus establishes that shape memory activation temperature is not, as plaintiffs suggest, a "functional" concept that "can vary as circumstances require."

Instead, the prosecution history reveals that the coinventor himself regarded shape memory activation temperature as a known temperature ascertainable by calculation and dependent upon a given thermoplastic's physical properties. Accordingly, the court shall construe the term, "shape memory activation temperature," as the '196 patent's coinventor understood it--namely, as the crystallization point of a given thermoplastic, calculated according to the following formula:

> T_c <<congruent>> (T_m + T_g)/2 where T_c is the crystallization temperature, T_m is the melting temperature and T_g is the glass transition temperature.

(D.I. 17, Ex. 3 at A363; *see also* A351 (equating shape memory activation temperature with a material's crystallization temperature)).

B. "Second Predetermined Pressure"

Elements (e)-(f) of Claim 12 disclose a two-step pressurization method designed to "conform the liner substantially precisely to the interior wall surface contours of the pipe." (D.I.17, Ex. 2, col.11, lns.33-35) Specifically, the '196 patent teaches

> pressurizing the interior of said liner to a first predetermined pressure above atmospheric pressure by means of said heated fluid and (3) reheating said liner to a predetermined temperature by heat transfer from said heated fluid to said liner, whereby, the liner returns substantially to its remembered cylindrical cross-section; and
>
> ***10** (f) then increasing the pressure in said liner to a second predetermined pressure above said first predetermined pressure....

(D.I.17, Ex. 2, col.11, lns.23-33) (emphasis added). Both parties agree that the claim calls for a two stage pressurization process. The first stage of pressurizing the interior of the liner must return the liner to its remembered cylindrical cross-section. The second stage of pressurization must "be above said first predetermined pressure to conform the liner substantially precisely to the interior wall surface contours of the pipe." (D.I.17, Ex. 2, col.11, lns.33-35) In the preferred embodiment, the pressure rises to about seven bars during the first pressurization stage and to about fifteen bars during the second pressurization stage. (D.I.17, Ex. 2, col.3, lns.31-32, 40) The claims and the specification reveal that both the first and second pressures must be "predetermined." Although the patent does not specifically define "predetermined," the court shall construe it, according to its ordinary meaning, as "determined beforehand." *Webster's Third New International Dictionary* 1786 (unabridged ed.1993). That is, the exact pressurization of the liner must be known before the pressurization process begins. Based on the plain meaning of the claim language, the second predetermined pressurization stage begins after the liner has returned "substantially to its remembered cylindrical cross-section." The court notes, however, that there is no limitation on how long the first predetermined pressure must be maintained after the rerounding of the liner and before the commencement of the second pressurization stage.

The court, therefore, shall construe the term "second predetermined pressure" to mean a pressure, determined before the second pressurization stage begins, which is above the first predetermined pressure and which conforms the liner substantially precisely to the interior of the host pipe.

C. The Cooling Step

Claim 12(h) discloses a step designed to set the liner in its final form within the host pipe. To this end, the claim teaches the following:

> (h) while the liner is still hot, introducing a cooling fluid into the liner for flow therethrough to fix the liner in final form in conformance to the interior wall of the pipe.

(D.I.17, Ex. 2, col.11, lns.36-col.12, ln.2) Defendant contends that this cooling stage must occur after the second pressurization stage. Plaintiffs, on the other hand, argue that the cooling stage can occur contemporaneously with the second pressurization stage.

As an initial matter, the court notes that the claim language itself indicates that the cooling stage disclosed in element (h) occurs after the second pressurization stage taught in element (f). The cooling stage described in element (h) is required to "fix the liner in final form in conformance to the interior wall of the pipe." (D.I. 17, Ex. 3, col. 11, ln. 37-col.12, lns. 1-2 (claim 12(h)) This "final form" cannot be achieved without "increasing the pressure in said liner to a second predetermined pressure ... to conform the liner substantially precisely to the interior wall surface contours of the pipe." (D.I. 17, Ex. 3, col. 11, lns. 31-35 (claim 12(f))

***11** The '196 patent's specification provides further support for construing claim 12 to require the cooling stage to occur after the second predetermined pressurization stage. After describing the second predetermined pressurization stage of the preferred embodiment, the specification states:

> Thereafter, valve 60 is closed, hot water supply 58 disconnected, and the hot water within the pipe is emptied. The packer/expander assemblies 52, 54 are then withdrawn. It is a further feature of this invention that, while the liner is still hot, a conventional expansion pig, having a diameter substantially identical to the inside diameter of the expanded liner, is introduced into the pipe 10 and is pushed through the pipe section applying a radial force to the liner so as to squeeze any remaining air from between the pipe liner and to thereby conform 100% of the liner surface against the interior surface of the pipe. The pig is preferably driven by a supply of cold water which more or less "freezes" the plastic into final form behind the pig, eliminating all air spaces between the liner and pipe section.

(D.I.17, Ex. 3, col.8, lns.11-25) (emphasis added). Thus, both the claim language itself and the specification contemplate a cooling stage after the second predetermined pressurization stage. Therefore, the court shall construe claim 12(h) as requiring the cooling stage to occur sometime after the second predetermined pressurization stage.

D. The Diameter of the Liner

Claim 12(a) provides "an elongate hollow liner ... having an original outer diameter substantially comparable to the inside diameter of the pipe to be lined...." (D.I.17, Ex. 3, col.10, lns.63-68) The parties dispute the scope of the term "substantially comparable to." Defendant contends in its claim construction briefs that "substantially comparable to" means the liner must be equal to or larger in diameter than the host pipe. (D.I. 121 at 22) Plaintiffs argue that "substantially comparable to" encompasses pipe liners "slightly less than, equal to, or slightly greater than the host pipes into which they are being installed." (D.I. 127 at 15)

The ordinary meaning of "substantially comparable to" the inside diameter of the pipe includes diameters slightly less than, equal to, or slightly greater than the host pipe diameter. "Substantial" is defined as "being that specified to a large degree or in the main." *Webster's* at 2280. A liner diameter that is "to a large degree" comparable to the diameter of a host pipe includes liner diameters that approximate the diameter of the host pipe. Thus, pipe liners with diameters that "substantially compare to" the diameters of their host pipes include diameters slightly smaller than, equal to, or slightly greater than the host pipe diameter.

Defendant, however, contends that the ordinary meaning of "substantially comparable to" does not control because the intrinsic evidence of the '196 patent is inconsistent with this meaning. Only two situations provide sufficient justification for defining a claim term in a manner other than its ordinary and accustomed meaning. *See* Johnson Worldwide

Assocs., Inc. v. Zebco Corp., 175 F.3d 985, 990 (Fed.Cir.1999). The first of those situations occurs when a patentee has chosen to be his or her own lexicographer by clearly setting forth an explicit definition for a claim term. *See id.* This is not the case here. The other situation occurs when the term or terms chosen by the patentee so deprive the claim of clarity that there is no means by which the scope of the claim may be ascertained from the language used. *See id.*

***12** Defendant urges that the intrinsic evidence of the '196 patent deprives the term "substantially comparable to" of its ordinary meaning. Defendant specifically points to the '365 patent specification, which explains that,

> [i]n practice, the liner configuration has an outside diameter equal to or slightly greater than the inside diameter of the pipe to be protected, whereby the said liner is either unstressed or under slight circumferential compression....

(D.I. 65, Tab 1, Ex. C., col. 1, lns. 40-47) (emphasis added). Defendant also notes that the '196 patent specification describes the preferred embodiment as having "a diameter slightly larger than the interior diameter of the pipe to be lined." (D.I.17, Ex. 3, col.1, lns.65-69) (emphasis added). Each of these specifications, however, describes the preferred embodiment, and a description of the preferred embodiment cannot limit a claim term. *See, e.g., Johnson Worldwide Assocs.,* 175 F.3d at 992. Moreover, defendant admitted in its reply brief in support of summary judgment that claim 12 "is broad and necessarily covers diameters that are smaller than the pipe to be lined." (D.I. 71 at 19) Accordingly, there is no compelling reason to deprive the term "substantially comparable to" of its ordinary meaning. The court shall construe the term "substantially comparable to" as encompassing pipe liners slightly less than, equal to, or slightly greater than the host pipes into which they are being installed.

With these constructions of the disputed claims in mind, the court now turns to defendant's motion for summary judgment.

IV. STANDARD OF REVIEW

A court shall grant summary judgment only if "the pleadings, depositions, answers to interrogatories, and admissions on file, together with the affidavits, if any, show that there is no genuine issue as to any material fact and that the moving party is entitled to judgment as a matter of law." Fed.R.Civ.P. 56(c). The moving party bears the burden of proving that no genuine issue of material fact exists. *See Matsushita Elec. Indus. Co. v. Zenith Radio Corp.,* 475 U.S. 574, 586 n. 10 (1986). "Facts that could alter the outcome are 'material,' and disputes are 'genuine' if evidence exists from which a rational person could conclude that the position of the person with the burden of proof on the disputed issue is correct." *Horowitz v. Federal Kemper Life Assurance Co.,* 57 F.3d 300, 302 n. 1 (3d Cir.1995) (internal citations omitted). If the moving party has demonstrated an absence of material fact, the nonmoving party then "must come forward with 'specific facts showing that there is a genuine issue for trial." ' *Matsushita,* 475 U.S. at 587 (quoting Fed.R.Civ.P. 56(e)). The court will "view the underlying facts and all reasonable inferences therefrom in the light most favorable to the party opposing the motion." *Pennsylvania Coal Ass'n v. Babbitt,* 63 F.3d 231, 236 (3d Cir.1995). The mere existence of some evidence in support of the nonmoving party, however, will not be sufficient for denial of a motion for summary judgment; there must be enough evidence to enable a jury reasonably to find for the nonmoving party on that issue. *See Anderson v. Liberty Lobby, Inc.,* 477 U.S. 242, 249 (1986). If the nonmoving party fails to make a sufficient showing on an essential element of its case with respect to which it has the burden of proof, the moving party is entitled to judgment as a matter of law. *See Celotex Corp. v. Catrett,* 477 U.S. 317, 322 (1986).

V. ANALYSIS

A. Literal Infringement

***13** Arguing that it does not deform its liner at the liner's shape memory activation temperature, defendant moves for summary judgment that its Sure-Line® process does not literally infringe claim 12 of the '196 patent. A finding of literal infringement requires that the asserted claims read on the accused process. *See* Morton Int'l. Inc. v. Cardinal Chem. Co., 5 F.3d 1464, 1468 (Fed.Cir.1993). A claim covers, or reads, on an accused process if every limitation recited in the claim is found in the accused process. *See* Kahn v. General Motors Corp., 135 F.3d 1472, 1477 (Fed.Cir.1998). Each limitation of the claim must be met exactly by the accused process, and any deviation from the claim precludes a finding of literal infringement. *See* Lantech, Inc. v. Keip Mach. Co., 32 F.3d 542, 547 (Fed.Cir.1994). If an express claim limitation is absent from an accused process, there can be no finding of literal infringement as a matter of law. *See* Kahn, 135 F.3d at 1477.

The court has construed shape memory activation temperature to mean the crystallization point of a given thermoplastic, calculated according to the following formula:

> T_c <<congruent>> $(T_m + T_g)/2$ where T_c is the crystallization temperature, T_m is the melting temperature and T_g is the glass transition temperature.

Viewed in light of this construction, the record evidence conclusively establishes that defendant's Sure-Line® process does not literally infringe claim 12(a)'s requirement that the pipe liner be deformed at a "shape memory activation temperature." Plaintiffs do not dispute that defendant deforms its liners at a temperature ranging between 120° F and 145° F. (D.I. 17 at A455-59; D.I. 65 at B120) Nor do plaintiffs dispute that defendant employs a high density polyethylene liner material with a crystallization point ranging between 126° C to 130° C (258.8° F to 266° F). (D.I. 17 at A455-59) Thus, there is no dispute that defendant's deformation temperature range of 120° F to 145° F falls far short of its liner's shape memory activation temperature range of 258.8° F to 266° F. Because an express claim limitation of the '196 patent is, therefore, absent from the accused Sure-Line® process, there can be no finding of literal infringement as a matter of law. *See* id. Accordingly, the court shall grant defendant's motion that it does not literally infringe the '196 patent.

B. Infringement Under the Doctrine of Equivalents

Although an accused device may not literally infringe a claim limitation, it may infringe under the doctrine of equivalents if "the differences between the claimed invention and the accused device ... are 'insubstantial.'" ' *Texas Instruments Inc. v. Cypress Semiconductor Corp.,* 90 F.3d 1558, 1563-64 (Fed.Cir.1996). A finding of insubstantiality turns on whether "the element of the accused device at issue performs substantially the same function, in substantially the same way, to achieve substantially the same result, as the limitation at issue in the claim." *Dawn Equip. Co. v. Kentucky Farms Inc.,* 140 F.3d 1009, 1016 (Fed.Cir.1998). Although the doctrine of equivalents "extends the protection of the patent beyond the literal words of the claims, it is not proper 'to erase a plethora of meaningful structural and functional limitations of the claim on which the public is entitled to rely in avoiding infringement.'" ' *Malta v. Schulmerich Carillons, Inc.,* 952 F.2d 1320, 1327 (Fed.Cir.1991) (quoting *Perkin-Elmer Corp. v. Westinghouse Elec. Corp.,* 822 F.2d 1528, 1532 (Fed.Cir.1987)). The United States Supreme Court has indicated that the particular linguistic framework used to test equivalency is not important, so long as it addresses the "essential inquiry [of whether] the accused product or process contain[s] elements identical or equivalent to each claimed element of the patented invention." *Warner-Jenkinson Co. v. Hilton Davis Chem. Co.,* 520 U.S. 17, 40 (1997). The Court emphasized that, "[t]he determination of equivalence should be applied as an objective inquiry on an element-by-element basis." *Id.* In the context of a summary judgment motion, "[w]here the evidence is such that no reasonable jury could determine two elements to be equivalent, district courts are obliged to grant partial or complete summary judgment." *Id.* at 39 n. 8.

1. "Shape Memory Activation Temperature"

***14** At issue is whether genuine issues of material fact exist with respect to whether defendant's method of deforming its pipe liner performs substantially the same function, in substantially the same way, to achieve substantially the same result as the process disclosed by element (a) of claim 12. *See, e.g.,* Dawn Equip. Co., 140 F.3d at 1016. The facts underlying this inquiry are not in dispute. The only question before the court concerns the scope of the claim term "shape memory activation temperature" and whether this term encompasses, by virtue of the doctrine of equivalents, temperatures manifestly below a given thermoplastic's shape memory activation temperature. Because this is a legal (as opposed to factual) dispute, it is amenable to resolution by the court on summary judgment.

From the evidence of record, it is apparent that defendant's Sure-Line® process performs the same function as that taught in element (a) of claim 12. Like element (a), the Sure-Line® process alters the cross-section of the thermoplastic pipe liner to enable the liner to be pulled into the host pipe. The way in which the Sure-Line® process performs this function, however, differs substantially from that disclosed by element (a) of claim 12. It is undisputed that defendant alters the cross-section of its thermoplastic liners by heating them to a range of 120° F to 145° F, which is substantially below the thermoplastic liner's shape memory activation temperature. Element (a) of claim 12, on the other hand, calls for alteration of the pipe liner's generally cylindrical cross-section "at a shape memory activation temperature." Thus, as the prosecution history conclusively establishes, the two processes employ significantly different temperatures in order to deform their liners. This difference in temperature produces substantially different results. A thermoplastic liner altered at a shape memory activation temperature "has substantially no tendency to return to its original shape once the deforming stresses are removed." (D.I. 17 at A356) On the other hand, a liner deformed according to the Sure-Line® process will reround after deformation unless restrained by tensile tape. (D.I. 65 at B058-059, B067-069) In short, element (a) teaches the use of a thermoplastic liner's memory activation properties to produce a liner that necessarily "has substantially no tendency to return to its original shape." Defendant's process, however, does not exploit the memory activation properties of thermoplastics and, as a consequence, its process requires the additional step of wrapping its deformed liners in tensile tape until the liner's insertion into the host pipe.

Thus, to ignore the significance of element (a)'s use of "shape memory activation temperature" and to conclude that the Sure-Line® process is the substantial equivalent of element (a) would allow element (a) "such broad play as to effectively eliminate that element in its entirety." *Warner-Jenkinson,* 520 U.S. at 29. Allowing the term "shape memory activation temperature" to encompass the temperatures at which defendant deforms its liners would erase a meaningful functional limitation of the claim upon which defendant was entitled to rely in avoiding infringement. *See Perkin-Elmer Corp.,* 822 F.2d at 1532. This conclusion is buttressed by the fact that plaintiffs distinguished the '196 patent from prior art by touting the patent's use of memory activation properties. For example, in distinguishing a prior art patent's ("the Steketee patent") use of PVC as a pipe liner, plaintiffs explained that, "PVC does not have shape memory characteristics, and, consequently, the cross section of the PVC liner in Steketee is not shaped ... with respect to any shape memory activation temperature at all." (D.I. 17 at A310) In light of the substantial difference between defendant's deformation process and that disclosed by element (a), no reasonable jury could find defendant's deformation process equivalent to element (a) of claim 12. Accordingly, the court shall grant defendant's motion for noninfringement of the '196 patent. In light of this conclusion, the court need not address defendant's other arguments in support of noninfringement under the doctrine of equivalents.

D. Defendant's Invalidity Argument

*15 Finally, defendant argues that claim 12 of the '196 patent omits an essential element of the invention and, therefore, is invalid for failure to comply with written description requirement of § 112, ¶ 1 of the patent act. *See* 35 U.S.C. § 112. Because courts presume patents are valid, the movant on summary judgment must establish invalidity by clear and convincing evidence. *See Electro Med. Sys. S.A. v. Cooper Life Sciences,* 34 F.3d 1048, 1052 (Fed.Cir.1994). Defendant contends that claim 12, which provides for a liner with a diameter "substantially comparable to" the inside diameter of the host pipe, includes liners with "smaller" diameters and, therefore, conflicts with the '196 patent's specification, which describes only liners with "larger" diameters. Although § 112 does not explicitly recognize an "omitted element" cause of action, the Federal Circuit appears to have so interpreted § 112. *See Gentry Gallery, Inc. v. Berkline Corp.,* 134 F.3d 1473, 1479 (Fed.Cir.1998); *see also Purdue Pharma, L.P. v. F.H. Faulding & Co.,* 48 F.Supp.2d 420, 427-28 (D.Del.1999); *Reiffin v. Microsoft Corp.,* No.C-98- 0266-VRW, 1998 WL 397915, at *3 (N.D.Cal. Jul. 10, 1998) (recognizing an "omitted element test"); Cindy I. Liu, Comment, 14 Berkeley Tech. L.J. 123 (1999) (analyzing the *Gentry Gallery* opinion).

In *Gentry Gallery,* an owner of a patent for a sectional sofa brought an infringement action against a competitor. The patent in suit disclosed a sectional sofa containing two parallel reclining seats separated by a fixed console, which the original disclosure of the invention described as containing the controls for reclining the seats. This original disclosure did not suggest any alternative location for the controls. *See Gentry Gallery,* 134 F.3d at 1479. Nonetheless, Gentry asserted subsequently added claims covering sofas in which the controls were not located on the console. Berkline argued that these subsequently added claims were invalid for omitting an essential element of the invention. In siding with Berkline, the Federal Circuit ruled that the patent owner could not assert claims that omit elements of the invention as originally disclosed, where one skilled in the art would recognize that the omitted element was essential to the invention as originally disclosed. *Id.* at 1479- 80; *accord Reiffin,* 1998 WL 397915, at *3. In applying this "omitted element" test, the Federal Circuit looked to the original disclosure of the invention, to the broadest original claim, and to the inventor's statements in the prosecution history. *See id.* at 1479. Each of these sources demonstrated that placement of the controls on the console was "the only possible location for the controls." *Id.*

In the present case, the original disclosure of the '196 patent described the invention as an improvement over the Laurent prior art, which utilized a liner with "an outside diameter ... approximately equal to the inside diameter of the pipe to be lined." (D.I. 17, Ex. 3 at A024) The improvement relates to, *inter alia,* "causing the liner to conform even more precisely to the interior contour of the pipe." (D.I. 17, Ex. 3 at A025-26) The disclosure explains that, "[t]o this end, thermoplastic material is extruded and calibrated to obtain a cylindrical insert liner with a diameter slightly larger than the interior diameter of the pipe to be lined." (D.I. 17, Ex. 3 at A026) At no point does the original disclosure (or, for that matter, the final version of the disclosure statement) declare that "larger" diameters are essential to the invention.

***16** Claim 1, as originally filed, disclosed no restrictions on the liner's diameter. (D.I. 17, Ex. 3 at A044) Moreover, the applicants' statements regarding the necessity for liners with diameters larger than the host pipe related to amendments to claim 1--amendments that added the "slightly larger than the interior diameter of the pipe to be lined" language. Plaintiffs also have submitted expert testimony that one skilled in the art would not regard a liner diameter larger than the host pipe's interior diameter as "essential" to the invention. (D.I. 67 at 20) For these reasons, defendant has failed to prove by clear and convincing evidence that claim 12 is invalid for omitting an essential element. Accordingly, the court shall deny defendant's motion for summary judgment that claim 12 is invalid.

VI. CONCLUSION

For the aforementioned reasons, the court shall grant defendant's motion that it does not infringe the '196 patent either literally or under the doctrine of equivalents. Defendant's motion is denied in all other respects. An appropriate order shall issue.

Not Reported in F.Supp.2d, 1999 WL 1011974 (D.Del.)

END OF DOCUMENT

JA2147 THROUGH JA 2158
REDACTED

ARTICLES

Introducing ActiveX

David Chappell - January 1997

Let's begin with a show of hands: how many of you hate Microsoft? And whether or not you hate Microsoft, how many of you believe that the company doesn't know the first thing about object technology, that, in fact, it has in some ways done damage to the entire concept of objects?

On the first point, I can't help you (although I might point out that letting your feelings about a company, especially one that's so unquestionably important, dominate your thinking is unlikely to lead to good decisions). The second point, though, while a widely held belief, is simply untrue. Microsoft's overarching approach to objects, embodied in the Component Object Model (COM), in fact hews much more closely to the standard conventions of objects than many people believe. And where it departs from those conventions, the departure results from COM's specific goals, not from the ignorance of its designers.

DavidChappe
& Associa

There is an area related to objects, however, where I believe it's quite justified to criticize Microsoft: terminology. Judging from the evidence, Microsoft doesn't seem to care much about the labels it assigns to things. Exhibit A here is the company's mutable names for COM-based technologies. The broad set of technologies that are built using COM were originally given the label "OLE", and Microsoft spent several years trying to convince us that OLE meant more than just Object Linking and Embedding, the compound documents technology from which the acronym was derived. In early 1996, however, Microsoft changed terminological direction, and the name "OLE" was deemed to once more refer only to compound documents. A new label, ActiveX, was introduced as an umbrella term for COM-based technologies, and several technologies under that umbrella were also renamed.

Defining ActiveX

It's just not possible to give a clear technical definition of what the term ActiveX means. The reason for this is simple: ActiveX is a marketing label, not a technical term. The clearest way to think about it is to view ActiveX as a brand name, like, say, Chevrolet. General Motors assigns the Chevrolet brand to a varied collection of cars, and what that collection includes changes over time. Still, all Chevrolets have some common elements, and assigning them all a common name makes GM's marketing task easier. Over the years, GM has built brand recognition and loyalty to Chevrolet in their customers' minds.

Microsoft is trying to do the same thing with ActiveX. There are many technologies grouped under this label, and exactly what those technologies are changes over time. Still, all of them have something in common--they all use COM--and Microsoft has been doing their level best to build equity in the form of customer recognition and loyalty around the ActiveX brand name.

It's tempting to define ActiveX as the set of all technologies that use COM. Sadly, this definition doesn't really work. The reason is that most Microsoft software today uses COM in some way, including Word, Excel, and even Windows 95 and Windows NT. Nobody is willing to argue that all of these products fall under the ActiveX umbrella.

When the term was first introduced in early 1996, it was used to refer to technologies that were somehow associated with the Internet and the World Wide Web, and the label still retains much of that flavor. Today, though, Microsoft includes COM itself as part of ActiveX, and there's nothing at all Internet-centric about COM. Once again, the only accurate perspective is to think of ActiveX as a flexible brand name. This generality notwithstanding, it's still possible to group today's most visible ActiveX technologies into four broad categories: fundamental technologies, component software support, middleware, and Web-related technologies.

Fundamental ActiveX Technologies

This first category includes COM itself, together with various COM-based building blocks. COM defines conventions and provides basic services for creating and working with software abstractions called COM objects. It's possible to implement COM objects in C++, Java, Delphi, and many other languages. Each COM object provides services to its clients through methods (of course), grouped together into one or more interfaces. COM objects exhibit the traditional characteristics of object technology: encapsulation, polymorphism, and even, counter to a widespread misconception, inheritance (although COM supports only interface inheritance, not inheritance of an object's actual implementation).

Other fundamental COM-based technologies include:

- monikers, a way to identify and instantiate specific object instances;

- support for object persistence;

- automation, which is really nothing more than another way for a COM object to expose its methods (if you're familiar with CORBA, automation is quite similar to CORBA's Dynamic Invocation Interface);

- type libraries, typically files containing a description of an object's interfaces (again, for the CORBA-literate, the idea is much like CORBA's Interface Repository);

- uniform data transfer, a standard way to move data between objects;

- connectable objects, standard interfaces for exchanging the information required to let an object talk back to its client.

Component Software Support

The second of our (somewhat arbitrary) categories is embodied in the idea of ActiveX Controls. Originally known as OLE Controls or OCXs, an ActiveX control is just an ordinary COM object that follows certain rules. To create reusable binary components, it's useful to define standard ways to do common things. For example, many objects will want to present their own user interface, send events, and expose their methods to other software. The ActiveX Controls specification defines standard interfaces for doing all of these things and more. To qualify as a control, a COM object must use the standard ActiveX Controls interfaces whenever it performs a function supported by those interfaces. For example, all controls must expose their methods via automation, and controls that provide a user interface must do so using the standard ActiveX Controls conventions.

Any piece of software that knows how to load and use ActiveX controls is an ActiveX control container. Popular ActiveX control containers include Visual Basic, PowerBuilder, and Microsoft's web browser, Internet Explorer. There is a large third-party market in this technology, with hundreds of companies building and selling reusable components packaged as ActiveX controls.

Middleware

Middleware is typically defined as protocols that provide generally useful services for a variety of applications. Today's CORBA-based products are middleware, as are many other offerings from many sources.

In its initial incarnation, COM had nothing to do with middleware-in fact, it defined no distributed capabilities at all. Although COM was designed from the start to support distribution, this capability didn't actually appear in products until the release of Distributed COM (DCOM) in mid-1996. Even when DCOM did appear, it initially ran only on Windows NT and, a few months later, Windows 95. DCOM won't be available on a range of non-Microsoft operating systems until later in 1997, and even then, it will be primarily Software AG, not Microsoft, who provides it.

DCOM is similar in many ways to CORBA-based products. Both provide the support required for communication between objects on different systems. CORBA calls this support an Object Request Broker, or ORB, while Microsoft just calls it DCOM. Whatever it's called, the functionality is quite
similar.

DCOM accomplishes remote method invocation using Microsoft's RPC protocol, which was itself borrowed from the Open Software Foundation's Distributed Computing Environment (DCE). DCOM also provides distributed security services, originally based on those provided with Windows NT. DCOM does not provide a directory service, although Microsoft promises to ship one "soon". Shipping a general-purpose directory has proven hard for the Redmond behemoth, but it's a safe bet that they'll eventually get something out there. Since the great majority of CORBA-based products don't have directories either, this isn't too much of a competitive deficit at the moment.

Web-Related Technology

The fourth category of ActiveX offerings is Web-related technology. Microsoft's web browser makes heavy use of COM, and as mentioned earlier, it's also an ActiveX control container. Using a web browser as a control container allows downloading ActiveX controls from a web server, then running them inside the browser. In this way, controls can function somewhat like Java applets, allowing a web page to load not just data, but also code.

There are some obvious differences between Java applets and ActiveX controls in this regard. For one thing, applets ship a machine-independent bytecode to the client, while controls transfer a machine-specific binary. Still, the kind of machine for which this binary is intended, a Windows/Intel box, is, well, fairly common, so building downloadable components as ActiveX controls isn't as limiting as it might initially seem. Applets and controls each have strengths and weaknesses, and each has a place in the world of downloadable components.

ActiveX Today

Microsoft's competitors are fond of saying that ActiveX is just a new label for OLE, a five year old technology. In the nefarious world of cross-vendor insults, this charge has a rare attribute: it's largely true. It's interesting that Microsoft's competitors see this as a negative, since one might imagine that most users would prefer stable, tested technologies. But whatever its history, ActiveX is clearly here to stay. The reason is simple: Microsoft says so and, like it or not, Microsoft owns large parts of the software world.

Back to Top.

Speaking :: Writing :: Consulting :: Blog :: Newsletter :: About :: Contact :: Home

United States Patent and Trademark Office

Home | Site Index | Search | FAQ | Glossary | Guides | Contacts | eBusiness | eBiz alerts | News | Help

Trademarks > Trademark Electronic Search System (TESS)

TESS was last updated on Fri Sep 7 04:09:21 EDT 2007

TESS HOME | NEW USER | STRUCTURED | FREE FORM | BROWSE DICT | SEARCH OG | BOTTOM | HELP

Logout Please logout when you are done to release system resources allocated for you.

Record 1 out of 1

TARR Status | ASSIGN Status | TDR | TTAB Status ***(Use the "Back" button of the Internet Browser to return to TESS)***

Typed Drawing

Word Mark	VISUAL BASIC
Goods and Services	IC 009. US 026 038. G & S: computer programs; namely, utility programs, language processors and interpreters, and documentation sold therewith as a unit. FIRST USE: 19910515. FIRST USE IN COMMERCE: 19910515
Mark Drawing Code	(1) TYPED DRAWING
Serial Number	74111826
Filing Date	November 1, 1990
Current Filing Basis	1A
Original Filing Basis	1B
Published for Opposition	July 28, 1992
Registration Number	**1787376**
Registration Date	August 10, 1993
Owner	(REGISTRANT) MICROSOFT CORPORATION CORPORATION DELAWARE One Microsoft Way Redmond WASHINGTON 980526399
Assignment Recorded	ASSIGNMENT RECORDED
Attorney of Record	WILLIAM O FERRON JR
Disclaimer	NO CLAIM IS MADE TO THE EXCLUSIVE RIGHT TO USE "BASIC" APART FROM THE MARK AS SHOWN
Type of Mark	TRADEMARK
Register	PRINCIPAL
Affidavit Text	SECT 15. SECT 8 (6-YR). SECTION 8(10-YR) 20030712.
Renewal	1ST RENEWAL 20030712
Live/Dead	LIVE

Indicator

TESS HOME | NEW USER | STRUCTURED | FREE FORM | BROWSE DICT | SEARCH OG | TOP | HELP

|.HOME |. SITE INDEX| SEARCH | eBUSINESS | HELP | PRIVACY POLICY

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:		)	
		)	
	Thomas D. Ashoff et al.	)	Examiner: Ali M. Mashaal
		)	
Serial No.:	09/495,157	)	Group Art Unit: 2136
		)	
Filed:	January 31, 2000	)	Docket: 105.201US1
		)	
For:	SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR AUTHENTICATING USERS USING A LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP) DIRECTORY SERVER		

APPELLANTS' BRIEF ON APPEAL UNDER 37 C.F.R. 41.37(c)

Mail Stop Appeal Brief- Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Brief is presented in support of the Notice of Appeal mailed July 16, 2004, from the final rejection of claims 1-17 of the above-identified application, as set forth in the Final Office Action mailed February 18, 2004. A copy of the claims being appealed is enclosed as Appendix I.

The Commissioner of Patents and Trademarks is hereby authorized to charge Deposit Account No. 19-0743 in the amount of 250.00 which represents the requisite fee set forth in 37 C.F.R. § 41.2(b)(2).

Appellants respectfully request consideration and reversal of the Examiner's rejections of pending claims 1-17.

APPELANTS' BRIEF ON APPEAL UNDER 37 C.F.R. 41.37(c)

TABLE OF CONTENTS

Page

1. REAL PARTY IN INTEREST

The Real Party in Interest of the above-captioned patent application is Secure Computing Corporation, the assignee of the application.

2. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences for the above-referenced patent application.

3. STATUS OF CLAIMS

The present application was filed on January 31, 2000 with claims 1-17. A non-final Office Action was mailed September 10, 2003. A Final Office Action was mailed February 18, 2004. Claims 1-17 (Appendix I, Claims) stand rejected under 35 U.S.C. §103(a), remain pending, and are the subject of the present appeal.

4. STATUS OF AMENDMENTS

Appellants have not filed an amendment subsequent to the mailing of the Final Office Action on February 18, 2004.

5. SUMMARY OF CLAIMED SUBJECT MATTER

According to one embodiment, the present invention relates to a system (such as reference numeral 300, Figure 3) for authorizing client (such as reference numeral 102a, Figure 3) access to a network resource (such as reference numeral 118, Figure 3). The system includes a server (such as reference numeral 106, Figure 3; page 7, lines 11-17) that has at least one directory (such as reference numeral 204, Figure 3; page 7, lines 11-17) that can be accessed using a network protocol. The directory is configured to store information concerning an entity's organization (see Figure 4 generally, and page 7, lines 11-17). The system also includes a firewall (such as reference numeral 110, Figure 3; page 9, lines 1-3) that is configured to intercept network resource requests from a plurality of client users. The firewall is operative to authorize a network resource request based upon a comparison (page 9, lines 10-19) of the contents of at least part of one or more entries in the at least one directory to an authorization filter (see page 11, lines 7-15 for discussion relating to authorization filters). The authorization filter is generated based on a directory schema that is predefined by the entity (page 12, line 14-21).

According to another embodiment, the present invention relates to an authentication method (the method is represented by reference numerals 302, 304, 306, 308, and 310 in Figure 3) executed by a firewall (such as reference numeral 110 in Figure 3). The method includes receiving a network authorization request (such as reference numeral 304, Figure 3; page 8, line 20-page 9, line 2) from a client user (such as reference numeral 102a in Figure 3; page 8, line 20-page 9, line 2). The method also includes querying (reference numeral 306, Figure 3; page 9, lines 10-19), using a network protocol, at least one directory (reference numeral 204, Figure 3; page 9, lines 10-19) that is configured to store information concerning an entity's organization (see Figure 4, generally; page 7, lines 11-17). The query is based upon an authorization filter (see page 11, lines 7-15 for discussion relating to authorization filters) that is generated based on a directory schema that is predefined by said entity (page 12, line 14-21). The method further includes determining, based on the results of the query (reference numeral 308, Figure 3; page 9, lines 10-19), whether the contents of at least part of one or more entries in said at least one directory satisfy the authorization filter (page 9, lines 10-19). Finally, the method includes permitting the network resource request through the firewall if the authorization filter is satisfied (page 9, lines 17-19).

According to another embodiment, the present invention relates to a program product for enabling a processor in a computer system to implement an authentication process (the process is represented by reference numerals 302, 304, 306, 308, and 310 in Figure 3). The program product includes a computer usable medium having computer readable program code embodied in the medium for causing a program to execute on the computer system. The program code includes a first computer readable program code for enabling the computer system (such as reference numeral 110, Figure 3; page 8, line 20-page 9, line 2) to receive a network request from a client user (such as reference numeral 102a, Figure 3; page 8, line 20-page 9, line 2). The program code also includes a second computer readable program code for enabling the computer system to query (reference numeral 306, Figure 3; page 9, lines 10-19), using a network protocol, at least one directory (reference numeral 204, Figure 3; page 9, lines 10-19) that is configured to store information concerning an entity's organization (see Figure 4, generally; page 7, lines 11-17). The query is based upon an authorization filter (see page 11, lines 7-15 for discussion relating to authorization filters) that is generated based on a directory schema that is predefined by said entity (page 12, line 14-21). The program code also includes a third computer readable program code for enabling the computer system to determine, based on the results of the query (reference numeral 308, Figure 3; page 9, lines 10-19), whether the contents of at least part of one or more entries in the at least one directory satisfy the authorization filter (page 9, lines 10-19). Finally, the program code also includes fourth computer readable program code for enabling the computer system to permit the network resource request through the firewall if the authorization filter is satisfied (page 9, lines 17-19).

This summary does not provide an exhaustive or exclusive view of the present subject matter, and Appellant refers to the appended claims and their legal equivalents for a complete statement of the invention.

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether independent claims 1-17 have been erroneously rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,131,120 to *Reid* in view of the *Microsoft Computer Dictionary* 1997, in further view of *Check Point Account Management Client*, Version 1.0, September 1998, and in view of other art relevant to the dependent.claims (not at issue herein).

7. ARGUMENT

A. The Law Applicable Under 35 U.S.C. §103

MPEP §2142 states the basic applicable law governing obviousness of claimed subject matter:

> To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

B. Introduction

Claims 1-17 each include a limitation requiring a comparison between data from a directory and an authorization filter. None of the prior art cited in the Office actions to date teach or suggest such a comparison. For this reason, claims 1-17 should have been allowed.

C. Appellants' Invention

Appellants describe, and claim in claims 1-17, a firewall that communicates with a directory hosted on a server to determine if a network resource access request should be authorized or denied. Figure 1 depicts an example of such a firewall. The example presented in Figure 1, and discussed throughout, is intended to generally familiarize the reader with Appellants' invention, and is not intended to be an exhaustive description.

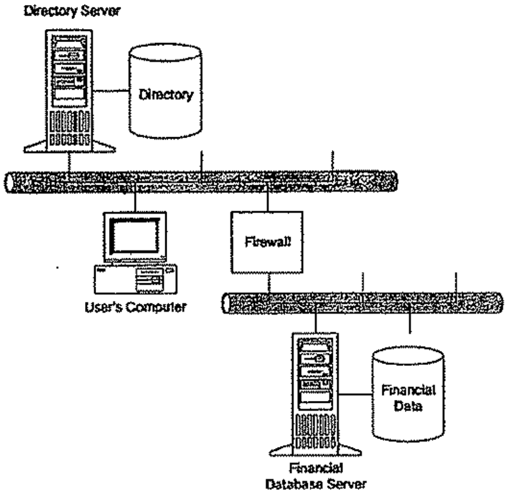


Figure 1

As can be seen from Figure 1, a firewall is interposed between user's computer and a financial database server, an example of a "network resource." Thus, any attempts to access the financial database server must pass through the firewall. The firewall intercepts the computer's first attempt to access the financial database server. See Application, page 9, lines 1-3.

In the wake of having intercepted the request, the firewall performs an operation to identify the user logged on to the computer from which the request emanated. See Application, page 9, lines 3-13. Based upon the identity of the user, the firewall queries a directory stored on a server to learn of attributes describing the user. Id.

A directory is similar to a database, in that it stores data that may be retrieved via query, but a directory is tailored to be read from more than it is written to. In the case of this invention, the directory stored on the server contains information concerning the organization of the entity employing the computer user. See Application, page 7, lines 11-17. For example, the directory may be organized as shown in Figure 2.

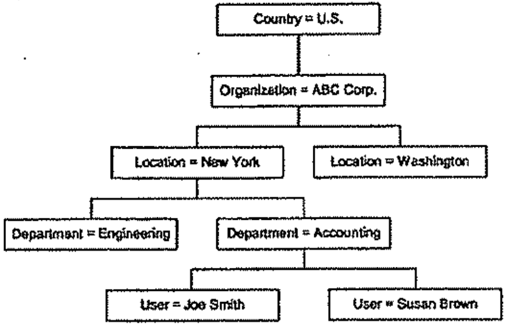


Figure 2

Thus, by virtue of querying such a directory with a given user's name, the firewall may obtain the department employing the user, the location in which the user is employed, the name of the organization employing the user, and the country in which the user is employed. Each of these pieces of information is an attribute describing the user.

After having obtained the user's attributes, the attributes are compared to an authorization filter. At its simplest, an authorization filter is an attribute and value pair. In the case of this example, the filter may be "Department = Accounting" (i.e., the user must work for the accounting department to be able access the financial database server). The firewall authorizes or denies the request on the basis of this comparison.

The act of comparing the authorization filter to data obtained from the directory is an element of every independent claim. For example, claim 1 requires "a comparison of the contents of at least part of one or more entries in said at least one directory to an authorization filter." Similarly, claims 8 and 17 require determining "whether the contents of at least part of one or more entries in said at least one directory satisfy said authorization filter." Thus, the independent claims require the general structure depicted in Figure 3, below

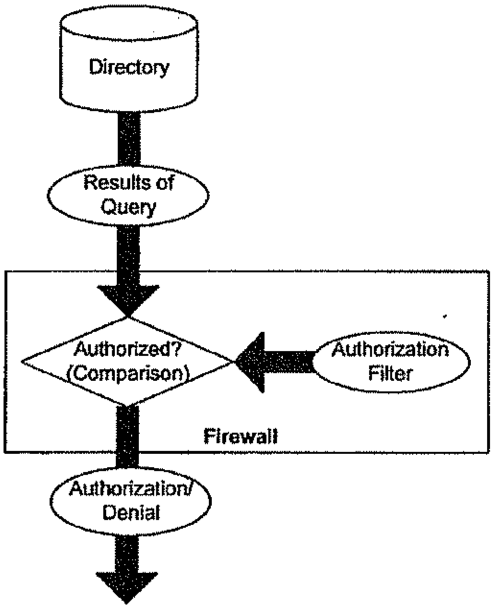


Figure 3

As shown in Figure 3, the claims require the authorization process to include a comparison of data returned from a query of the directory to an authorization filter.

D. The Prior Art

1. Summary of the Cited Prior Art

The prior art cited against the independent claims includes: (1) U.S. Patent No. 6,131,120 to *Reid*; (2) *Microsoft Computer Dictionary* 1997; and (3) *Check Point Account Management Client*, Version 1.0, September 1998. Of these, only the teachings of *Reid* are at issue. Briefly, *Microsoft Computer Dictionary* is cited to for its definition of the term "firewall," in order to support the proposition that the routers in *Reid* may be thought of as firewalls, because of the functionality they provide (Appellants do not dispute this). *Check Point Account Management Client* is a software manual, and is cited to support the proposition that it would have been obvious to modify the directory described in *Reid* to store information concerning an entity's organization (for the purposes of this appeal only, Appellants do not dispute this).

2. U.S. Patent No. 6,131,120 to *Reid*

Reid teaches a system whereby each router or gateway (collectively referred to herein as a router) stores a router access list. A router access list is a list of names and addresses of users and the destinations they are permitted to access. See Final Office Action, page 2 ("Each RAL [router access list] . . . includes the names and addresses of users and the destinations they are allowed to reach."). According to the scheme taught in *Reid*, a router intercepts a network resource access request, and extracts the requesting computer's address and requested destination address. If the router access list indicates that the requesting computer is permitted to access the requested address, the access is authorized, otherwise it is denied. See Final Office Action, pages 2-3.

The router access list taught by *Reid* is created automatically by a software application running on a server that stores a directory. See *Reid*, col. 8, lines 23-28. The directory contains, amongst other entries, the name and access privileges assigned to each user. See *Reid*, col. 8, lines 6-11 and 23-27. For each router in an organization's system, the application creates an appropriate router access list and sends the list to the router. See *Reid*, col. 8, lines 21-34.

The general structure of the authorization process described by *Reid* is depicted in Figure 4, below.

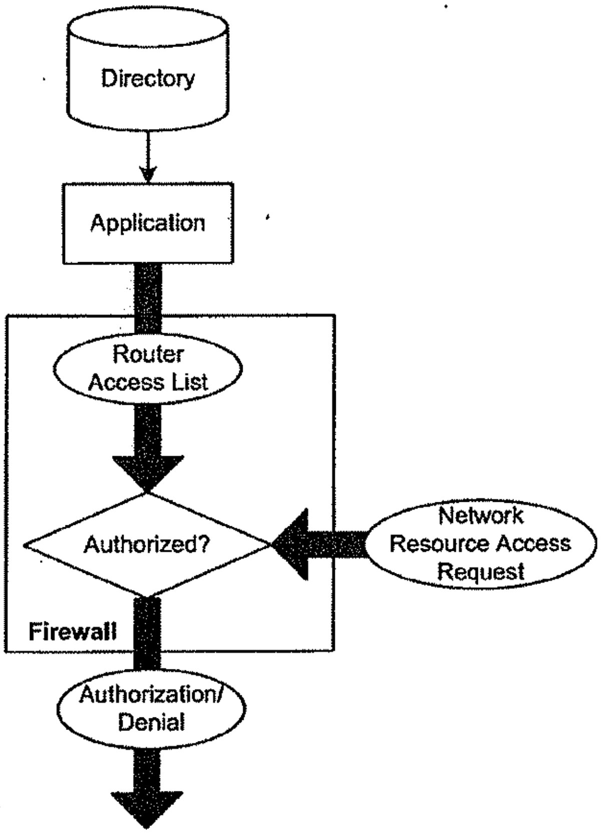


Figure 4

E. Brief Summary of Prosecution

The above-captioned application was filed on January 31, 2000. A first Office action was mailed on September 10, 2003 (claims 1-17 were rejected under 35 U.S.C. §103(a) as being obvious). Appellants filed a response on January 16, 2004. A final Office action was mailed on February 18, 2004 (claims 1-17 remained rejected under 35 U.S.C. §103(a), although the premise of the rejection changed). Finally, Appellants filed a notice of the present appeal on July 16, 2004.

F. Appellants' Rebuttal of the Rejections of the Independent Claims

1. The Initial Office Action

In a first Office action mailed September 10, 2003, the independent claims were rejected as being obvious in light of *Reid* and other prior art, the teachings of which are not presently at issue.

As stated above, each of the independent claims requires the act of comparing an authorization filter to data obtained from the directory. On its face, *Reid* wholly lacks any teaching or suggestion of such an act. To formulate his initial rejection, the Examiner seized upon an incorrect interpretation of *Reid*. Specifically, the Examiner assumed the existence of two separate unarticulated steps in Reid: (1) that the firewall transmitted access criteria, which the Examiner likened to an authorization filter, to the directory; and (2) that a comparison between the access criteria and the contents of the directory was performed as a part of the process of producing the router access lists. According to the Examiner,

> . . . in order for the directory to generate the appropriate [router] access list, each router/gateway must have transmitted its access criteria to the directory. The examiner further asserts that this criteria is an authorization filter and that in order for the directory to send back a correct access list, some comparison must have been made with directory entries and the router/gateway criteria (authorization filter).
>
> Initial Office Action, page 5.

Thus, according to the Examiner, the authorization process of *Reid* includes the extra—albeit completely unmentioned—steps, shown in dashed lines, depicted in Figure 5, below.

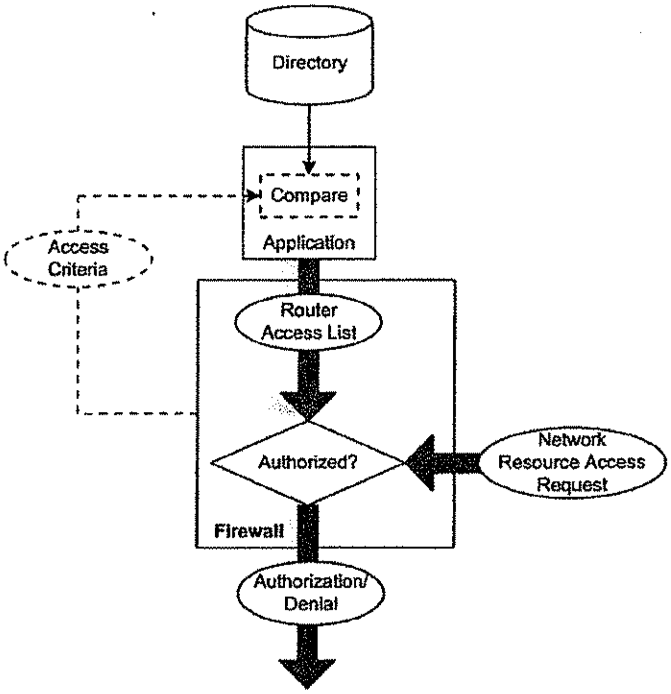


Figure 5

In a response filed January 16, 2004, Appellants argued that neither the "access criteria," nor the comparison step are referred to, or even hinted at, by the text of *Reid*. Further, Appellants went on to argue that the text of *Reid* plainly contradicts the very notion of their existence. Appellants argued that the router access lists of *Reid* are generated on the sole basis of the contents of the directory. Put simply, the application reads the directory and creates the router access lists. The process involves no transmission of access criteria, nor does it involve a comparison between access criteria and the contents of the directory.

In support of their position, Appellants cited column 6, lines 23-25 of *Reid* (emphasis added), which states that entries in the directory control the entire network:

> Because the directory knows the location and IP address of each user, and the location and IP address of each router/gateway, a directory application can periodically populate the RAL [router access list] in each router/gateway on the network using LDAP. Entries in the directory thereby control the entire network and the network router/gateway configuration management is automated.

The Examiner seems to have accepted this argument, because, in authoring his final rejection, he has altered his interpretation of *Reid*, in order to find a new way in which one might understand it to include a comparison between an authorization filter and data obtained from the directory.

2. The Final Office Action

In authoring his final rejection, the Examiner has reinterpreted *Reid* to find the missing element in a routine act performed by virtually every router that has been configured to act as a firewall: the act of comparing a network access request to a router access list. According to the Examiner, the routers of *Reid*

> compare[] the requesting device's address and requested destination to that information in the router/gateway which was provided by the directory server [i.e., the router access list], in order to determine whether the requesting device should be allowed/denied access. Therefore, the router/gateway clearly contains an authorization filter by which it can make a comparison of the content of at least

> part of one or more entries in the directory to determine which traffic may be allowed to pass through to a given destination.
>
> Final Office Action, pages 2-3.

The Examiner's latest interpretation of *Reid* still falls short of requirements posed by each of Appellants' claims. The independent claims require a comparison between an authorization filter and at least a portion of an entry in the directory. For example claims 1-7 require "a comparison of the contents of at least part of one or more entries in said at least one directory to an authorization filter." (Claims 8 through 17 include similar language.) The Examiner's interpretation does not even purport to find such a comparison in *Reid*. Instead, the Examiner's interpretation of *Reid* yields a comparison between a router access list (which the Examiner likens to an authorization filter) and address data extracted from the network access request. The difference between the claimed invention and the Examiner's latest interpretation of Reid is depicted in Figure 6.

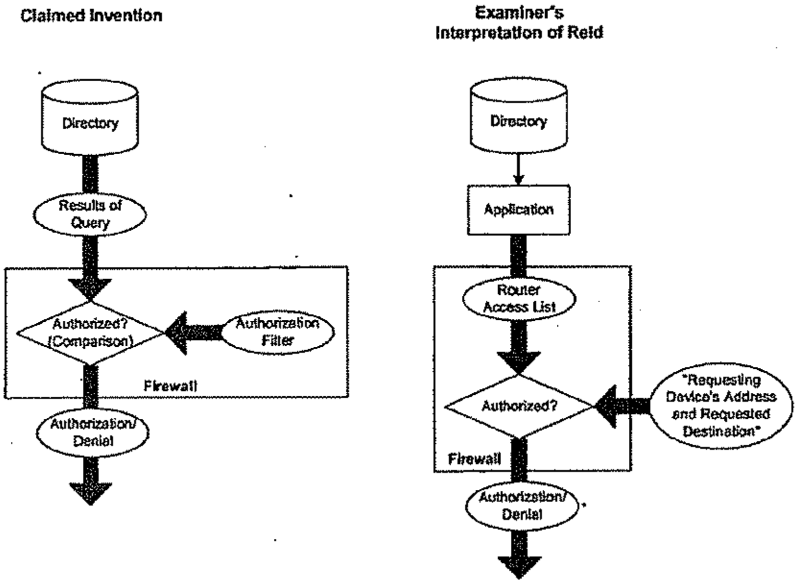


Figure 6

As can be seen from Figure 6, the claimed invention includes a comparison between an authorization filter and at least a portion of an entry from the directory (i.e. the "results of query" shown in Figure 6). Turning to *Reid*, on the other hand, even if one accepts the Examiner's position that a router access list is an authorization filter, the comparison does not occur between the proper subjects. Therein, at stated in the Examiner's Response to Arguments, the comparison occurs between the router access list (asserted to be an authorization filter) and "the requesting device's address and requested destination." See Final Office Action, page 2. The requesting device's address and requested destination are units of information extracted from the network resource access request—not from the directory. Therefore, at best, the comparison in *Reid* occurs between an authorization filter (router access list) and information extracted from the network resource access request[1], not the query results as described by Appellants and stated in claims 1-17. Plainly, such a comparison does not rise to textual requirements presented in claims 1-17.

[1] For the present purposes, Appellants take no position regarding whether a router access list may be properly characterized as an authorization filter. To the extent Appellants have appeared to acquiesce in this portrayal, Appellants have done so in order to present the Examiner's interpretation of *Reid*—not to endorse the Examiner's interpretation.

G. Conclusion

The prior art cited against claims 1-17 fails to teach or suggest a comparison between an authorization filter and data obtained from a directory. This act is required by each of the rejected claims. The record fails to present any motivation for one to modify *Reid* to include such a step. Indeed, it is not clear how *Reid* could so be modified, without altering the theory of operation of the firewall function presented therein. For at least this reason, the rejection of claims 1-17 is improper, and the rejection should be withdrawn.

Respectfully submitted,

THOMAS D. ASHOFF ET AL.

By their Representatives,

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. Box 2938
Minneapolis, MN 55402

Date 22 Feb. 2004 By

Nicholas P. Johns
Reg. No. 48,995

<u>CERTIFICATE UNDER 37 CFR 1.8:</u> The undersigned hereby certifies that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: Mail Stop Appeal Brief, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on this 22 day of February, 2005.

Peter Rebuffoni
Name

Signature

APPENDIX I

THE CLAIMS ON APPEAL

1. (Previously Presented) A system for authorizing client access to a network resource, comprising:

a server having at least one directory that can be accessed using a network protocol, said at least one directory being configured to store information concerning an entity's organization; and

a firewall that is configured to intercept network resource requests from a plurality of client users, said firewall being operative to authorize a network resource request based upon a comparison of the contents of at least part of one or more entries in said at least one directory to an authorization filter, wherein said authorization filter is generated based on a directory schema that is predefined by said entity.

2. (Original) The system of claim 1, wherein said at least one directory is a lightweight directory access protocol directory.

3. (Original) The system of claim 1, wherein said authorization filter is specified using a graphical user interface.

4. (Original) The system of claim 1, wherein said authorization filter implements a per-user authentication scheme.

5. (Original) The system of claim 1, wherein said authorization filter implements a per-service authentication scheme.

6. (Original) The system of claim 1, wherein said firewall and said directory communicate using secure socket layer communication.

7. (Original) The system of claim 1, wherein said firewall is configured to query multiple directories.

8. (Original) An authentication method at a firewall, comprising the steps of:

(a) receiving a network resource request from a client user;

(b) querying, using a network protocol, at least one directory that is configured to store information concerning an entity's organization, wherein said query is based upon an authorization filter that is generated based on a directory schema that is predefined by said entity;

(c) determining, based on the results of said query, whether the contents of at least part of one or more entries in said at least one directory satisfy said authorization filter; and

(d) permitting said network resource request through said firewall if said authorization filter is satisfied.

9. (Original) The method of claim 8, wherein step (b) comprises the step of querying said at least one directory using a lightweight directory access protocol.

10. (Original) The method of claim 8, further comprising the step of specifying an authorization filter using a graphical user interface.

11. (Original) The method of claim 10, wherein said specifying step comprises the step of specifying an authorization filter that implements a per-user authentication scheme.

12. (Original) The method of claim 10, wherein said specifying step comprises the step of specifying an authorization filter that implements a per-service authentication scheme.

13. (Original) The method of claim 8, wherein step (b) comprises the step of querying said directory using secure socket layer communication.

14. (Original) The method of claim 8, wherein step (b) comprises the step of querying multiple directories.

15. (Original) The method of claim 8, wherein step (a) comprises the step of receiving a network resource request from a client user at an internal network.

16. (Original) The method of claim 8, wherein step (a) comprises the step of receiving a network resource request from a client user at an external network.

17. (Original) A computer program product for enabling a processor in a computer system to implement an authentication process, said computer program product comprising:

a computer usable medium having computer readable program code embodied in said medium for causing a program to execute on the computer system, said computer readable program code comprising:

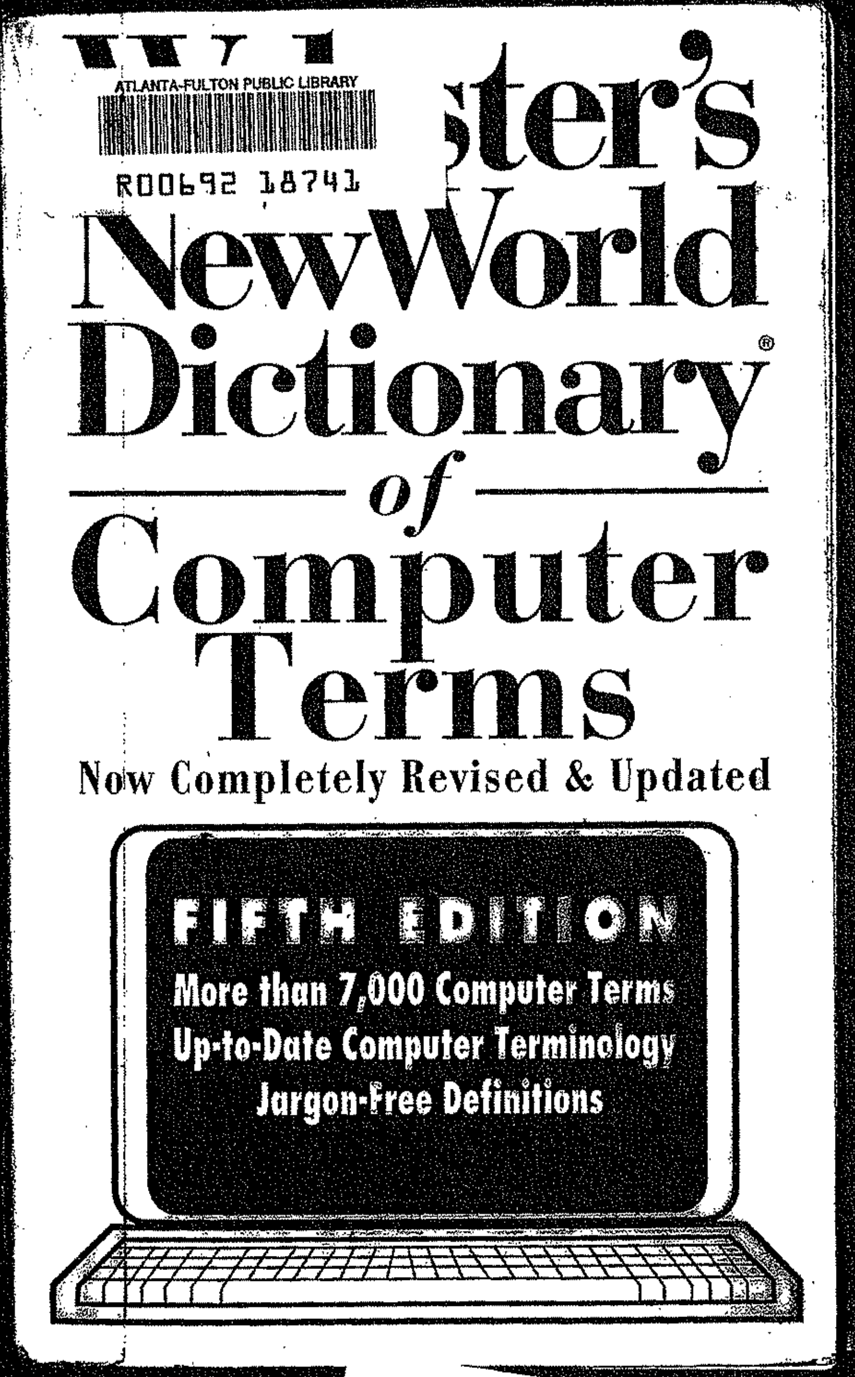
first computer readable program code for enabling the computer system to receive a network resource request from a client user;

second computer readable program code for enabling the computer system to query, using a network protocol, at least one directory that is configured to store information concerning an entity's organization, wherein said query is based upon an authorization filter that is generated based on a directory schema that is predefined by said entity;

third computer readable program code for enabling the computer system to determine, based on the results of said query, whether the contents of at least part of one or more entries in said at least one directory satisfy said authorization filter; and

fourth computer readable program code for enabling the computer system to permit said network resource request through said firewall if said authorization filter is satisfied.

JA2191 THROUGH JA2347 REDACTED



ster's NewWorld Dictionary® of Computer Terms

Now Completely Revised & Updated

FIFTH EDITION

More than 7,000 Computer Terms
Up-to-Date Computer Terminology
Jargon-Free Definitions

Macmillan General Reference
A Prentice Hall Macmillan Company
15 Columbus Circle
New York, NY 10023

A Webster's New World™ Book

MACMILLAN is a registered trademark of Macmillan, Inc.
WEBSTER'S NEW WORLD DICTIONARY is a registered trademark of Simon & Schuster, Inc.

Prentice Hall is a registered trademark of Prentice-Hall, Inc.

Dictionary Editorial Offices:
New World Dictionaries
850 Euclid Avenue
Cleveland, Ohio 44114

Library of Congress Cataloging-in-Publication Data

Spencer, Donald.

ISBN: 0-671-89993-7

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Fifth Edition

BHB

Introduction

It is unlikely that any field has contributed more new terms (or ne meanings of old ones) to the language in the last few years than h: computer technology. This is largely because computer technology is itse a new and changing field. As it evolves, fresh terminology must be develope to communicate, describe, and define the heretofore unknown concept components, and techniques. This book provides a glossary of over 7,0C words, phrases, acronyms, and abbreviations used in connection wit computers.

When a business installs its first computer, not only are the owners an managers apprehensive; the majority of the office and administrative pe sonnel are also involved in the change that has taken place. Managers mu know something about the current computer jargon in order to assist th staff with knowledgeable explanations of what is happening. Filling the nee for this information is the purpose of this book.

This new edition offers up-to-date coverage of terms used in reference t hardware, software, programming, logic, and computer graphics, as well a those used in ancillary fields such as data communications and artificia intelligence. Areas of frequent application are also covered, particularl mathematics and business administration.

The terms in this book have been selected as those most likely to confron the beginning computer-user, and they are defined in a way that any la person can understand. Wherever possible, technical terms have beer avoided so that the definitions might be easily read and understood. Wher

scheduled report A report produced at regular intervals to provide routine information to users.

scheduled maintenance Maintenance of a computer system at fixed intervals to maintain its reliability.

scheduler A program that schedules jobs for processing.

scheduling **(1)** The task of determining what the succession of programs should be in a multiprogramming computer center. **(2)** Allocating a nonsharable resource, such as CPU time or an I/O device, to a particular task for a period of time.

schema A structure for organizing knowledge relative to context or expectations; the definition of an entire database.

schematic symbols The symbols used in schematic diagrams.

scientific applications Tasks that are traditionally numerically oriented and often require advanced engineering, mathematical, or scientific capabilities. They seldom require the extensive file-handling capabilities of business applications.

scientific computer A computer that is capable of high-speed mathematical processing or "number crunching." See SUPERCOMPUTER.

scientific notation Notation in which numbers are written as a "significant digits" part, or mantissa, times an appropriate power of 10, or exponent.

scientific programming language A programming language that is designed to handle mathematical formulas and matrices, such as BASIC, FORTRAN, PASCAL, C, or MODULA-2.

scientific visualization A term describing technology that enables scientists to store vast amounts of mathematical data, generate graphical models that represent that data, and visually analyze the results, usually through interactive software programs. Scientific visualization is a multidisciplinary methodology that employs the largely independent but converging fields of computer graphics, image processing, computer vision, signal processing, and computer-aided design. Its specific goal is to act as a catalyst between scientific computation and scientific insight. Scientific visualization came into being to meet the ever-increasing need to deal with highly active, very dense data sources, including, for example, satellite data and data from supercomputer computations.

scissoring Automatic erasing of all portions of a design on the visual display device that lie outside user-specified boundaries.

SCM Abbreviation for Society for Computer Medicine, an organization that brings together physicians and computer scientists, emphasizing the use of automation for medical applications.

scope **(1)** The range of control of a software program. **(2)** An OSCILLOSCOPE.

SCR Abbreviation for SILICON-CONTROLLED RECTIFIER, a semiconductor device useful in controlling large amounts of DC current or voltage.

scrambling Similar to encryption, in which data or transmissions are "scrambled" so that they can only be retrieved by authorized users.

scrapbook A storage location for frequently used text and pictures. The stored images can be inserted into new documents as required.

scratch To delete data from memory.

scratch file A temporary file created during the processing of substantial files of data by copying all or part of a data set to an auxiliary storage device.

scratchpad Small, fast storage used in some computers in place of registers. Also called cache memory.

screen **(1)** A television-like output device that can display information. **(2)** A pattern of tiny dots used as shading in a graphic.

screen angle The angle at which a HALFTONE screen is printed.

screen capture **(1)** The transfer of the image on the current display screen into a graphics file. **(2)** A printout of the current screen display.

GARNER'S MODERN AMERICAN USAGE

Bryan A. Garner

OXFORD
UNIVERSITY PRESS
2003

OXFORD
UNIVERSITY PRESS

Oxford New York
Auckland Bangkok Buenos Aires Cape Town Chennai
Dar es Salaam Delhi Hong Kong Istanbul Karachi Kolkata
Kuala Lumpur Madrid Melbourne Mexico City Mumbai Nairobi
São Paulo Shanghai Taipei Tokyo Toronto

Published by Oxford University Press, Inc.
198 Madison Avenue, New York, New York 10016

www.oup.com

Oxford is a registered trademark of Oxford University Press

Library of Congress Cataloguing-in-Publication Data is available
ISBN 0-19-516191-2

9 8 7 6 5 4 3 2 1

Printed in the United States of America
on acid-free paper

[illegible] in the waves." Doug McCash, "3 [illegible] Venture Out of the Gallery Mainstream," [illegible] *Picayune* (New Orleans), 1 June 2001, Lagniappe §, at 36.

The adjective *plumb*, "perfectly straight, vertical," by dialectal extension of the "perfectly straight sense," has come to mean "entirely, wholly" <I'm plumb tired>. But some writers confuse the spelling by associating it somehow with fruit—e.g.: "Shelley, a 13-week-old springer spaniel, looks *plum-tired* [read *plumb tired*] during an obedience class at Temple Terrace Recreation Center on Tuesday night." "Inside," *St. Petersburg Times*, 16 Jan. 1993, Community Times §, at 1 (photo caption). See DIALECT.

Plumb is also a verb meaning "to measure depth, esp. of water." The confusion with *plum* occasionally persists with this sense as well—e.g.: "Her poetry is insightful in a way you might expect from someone who *plums the depths* [read *plumbs the depths*] of emotions and the mind." Paula Wachowiak, "For Masters of Verse, It's All Work and Word Play," *Buffalo News*, 6 Sept. 2000, at D1.

plurality. See **majority (C).**

PLURAL POSSESSIVES. See POSSESSIVES (B).

PLURALS. A. Generally. Most nouns form their plurals simply by adding *-s*—thus *books, songs, xylophones*. But if a word ends with the sound of *-s-*, *-sh-*, *-ch-*, or *-z-*, the plural is formed by adding *-es*—thus *buses, thrushes, churches*, and *buzzes*. Occasionally, a single final consonant is doubled—thus *fez* makes *fezzes*.

Several exceptions exist in words derived from Old English, such as *child–children, ox–oxen, man–men, woman–women, mouse–mice, louse–lice, foot–feet, goose–geese, tooth–teeth*.

B. Borrowed Words. References to this subentry appear throughout this book. That's not to say that each such term is elaborated on here but only that the principles governing the words are explained here.

Words imported into the English language from other languages—especially Greek, Latin, French, and Italian—present some of the most troublesome aspects of English plurals. Many imported words become thoroughly naturalized; if so, they take an English plural. But if a word of Latin and Greek origin is relatively rare in English—or if the foreign plural became established in English long ago—then it typically takes its foreign plural.

One reliable guide is this: if in doubt, use the native-English plural ending in *-s*. That way, you'll avoid the mistakes involved in HYPERCORRECTION, which is rampant with false foreign plurals (as when people say or write *ignorami* instead of *ignoramuses*, thereby betraying something quite ironic). H.W. Fowler called the benighted stab at correctness "out of the frying-pan into the fire" (*MEU1* at 416). Many writers who try to be sophisticated in their use of language make mistakes such as *ignorami* and *octopi*—unaware that neither is a Latin noun that, when inflected as a plural, becomes *-i*. The proper plural of the Greek word *octopus* is *octopodes*; the proper English plural is *octopuses*.

Those who affect this sort of sophistication may face embarrassing stumbles—e.g.: "A 'big city' paper with an editor as eminently qualified as I'm sure you are should know that the plural of *campus* is *campi* (not *campuses*). Just like the plural of *virus* is *viri* (not *viruses*), and the plural of *stadium* is *stadia* (not *stadiums*)." Letter to the Editor, *Dallas Morning News*, 22 Sept. 2002, at J3 (name withheld for obvious reasons). Although *stadia* has some basis as a plural in English (see **stadium**), *campi* and *viri* are ludicrous, and this attempted comeuppance reeks of ignorance.

Again, if it's a close call, use the native plural. In music, it's better to say *allegros* than *allegri*; *concertos* than *concerti*; *contraltos* than *contralti*; *solos* than *soli*; *sopranos* than *soprani*; and *virtuosos* than *virtuosi*. In publishing, it's better to say *appendixes* than *appendices*; *compendiums* than *compendia*; *Festschrifts* than *Festschriften* (from German); and *thesauruses* than *thesauri*. It's pedantic and prissy to say that politicians attend *fora*, enter *auditoria*, ascend *rostra*, and speak in favor of *referenda*.

But exceptions certainly exist. Literate people say *crises*, not *crisises*; *criteria*, not *criterions*; *hypotheses*, not *hypothesises*; *phenomena*, not *phenomenons*; and *timpani*, not *timpanos*. Medical and biological researchers say *bacilli*, not *bacilluses*; *fungi*, not *funguses*; *ova*, not *ovums*; *stimuli*, not *stimuluses*; and *thalami*, not *thalamuses*. Few other people ever use those words, although *fungi* and *stimuli* aren't uncommon.

Some are extremely close calls, or vary according to context. *Cactuses* predominates in common usage, but *cacti* is the more frequent form in botanical contexts. *Formulas* is generally better than *formulae* (and *spectrums* than *spectra*), but not in scientific contexts. There is significant movement toward *honorariums*, but *honoraria* still predominates; the same is true of *penumbras* and *penumbrae*. *Millenniums* and *millennia* are neck-and-neck, the former predominating in BrE and the latter in AmE.

And some variant forms have started undergoing DIFFERENTIATION. *Phalanxes* is the plural referring to groups of people; *phalanges* is the term for bones in the fingers and toes. *Protozoans* is the term for a few microorganisms that go by that name, but *protozoa* is typically used for large numbers. *Staff* generally makes *staffs*, but the musical staff is pluralized *staves*. *Stigmas* is better than *stigmata* except in religious contexts <the stigmata of Christ>, and *dogmas* is better than *dogmata*.

French words also present problems. *Bête*

SECURE COMPUTING

+1.40

Select

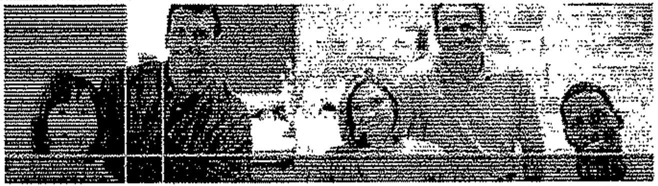
Resources Products Try/Buy Support Partners Company

Locator

Printable Version

Home > Company > About Secure >> Company Fact Sheet

Web Gateway | Messaging Gateway | Network Gateway



Company

About Secure

Company Fact Sheet

Executive Management

Board of Directors

Privacy Policy

Disclaimer

Press Center

Events / Seminars

Investor Relations

Careers

Contact Us

Company Fact Sheet

Company overview

Secure Computing® is a global leader in Enterprise Gateway Security Solutions. Powered by our **TrustedSource™ technology**, our best-of-breed portfolio of solutions provides **Web Gateway**, **Messaging Gateway**, and **Network Gateway** security, as well as **Identity and Access Management**. Secure Computing is proud to be the security solutions provider to many of the most mission-critical and sensitive environments in the world. Please see below for more information on our solutions.

Our more than 19,000 customers, supported by a worldwide network of partners, in Dow Jones Global 50 and the most prominent organizations in banking, financial se telecommunications, manufacturing, public utilities, and federal and local governme employees, the Company is headquartered in San Jose, California, and has offices

Financial status	**Market opportunity**	**Customers**
Patents	**Management team**	**Milestones**

Financial status

Secure Computing is publicly traded on the Nasdaq national Market System under revenues (reclassified for discontinued operations of the Advanced Technology cor

- 2006 Fiscal Revenues: $176.7M
- 2005 Fiscal Revenues: $109.2M
- 2004 Fiscal Revenues: $93.40M
- 2003 Fiscal Revenues: $76.21M
- 2002 Fiscal Revenues: $61.96M
- 2001 Fiscal Revenues: $48.35M
- 2000 Fiscal Revenues: $34.64M
- 1999 Fiscal Revenues: $22.54M

Market opportunity

Organizations today are expanding their businesses through the Internet daily. In tr environment, threats are also increasing right along with growth opportunities. Indu: expects the worldwide revenue for security hardware and software to be $30 billion

Our customers need a secure infrastructure they can rely on. Accordingly, our comi risk exposure and protect their information assets from a multitude of threats, incluc intruders, legal liability, security compromises, hackers, malicious software, and vir

Customers

Secure Computing's customers operate some of the largest and most sensitive net the world. They include the majority of the Dow Jones Global 50 Titans and numerc Fortune 1000, as well as banking, financial services, healthcare, telecommunicatior utilities, schools, and federal and local governments. Secure Computing has close largest agencies of the United States government, including multiple contracts for a research. Overseas, our customers are concentrated primarily in Europe, Japan, Cl Latin America.

Partners

Our partnerships include a global network of OEMs, members of our Secure Allianc systems integrators, and companies that include our solutions in their product offeri extensive support through our PartnersFirst Program and Web site. All business ex key accounts are sold through our partners. These companies include, for example Alternative Technology, Blue Coat Systems, Cisco, Computer Associates, Comstor F5, Hewlett-Packard, McAfee, Microsoft, Network Appliance, NetOne Systems, Nor PGP Corporation, SafeNet, Sun PS, SAIC, Tech Data, Vertex Link, Voltage Securit Westcon, and Workshare.

Patents

Secure Computing is a leader in advanced research and development of network a technology. Our team of distinguished researchers and scientists has achieved nun the security industry over the past 16 years. The company has been granted a total pending in the United States. These patents cover systems architecture, cryptograr filtering, and security control systems.

Management team

John McNulty — President, Chairman and Chief Executive Officer
Jay Chaudhry — Vice Chairman and Chief Strategy Officer
Tim Steinkopf — Senior Vice President and Chief Financial Officer
Vincent M. Schiavo — Senior Vice President, Worldwide Sales and Marketing
Mike Gallagher — Senior Vice President, Product Development and Support
Mary K. Budge — Senior Vice President, Secretary, and General Counsel
Dr. Paul Judge — Chief Technical Officer
Atri Chatterjee — Senior Vice President, Marketing

Secure Computing and Enterprise Gateway Security solutions

Secure Computing has long been regarded as the "gold standard" in Enterprise fire award-winning products protect some of the most mission-critical networks and apr these network environments have evolved with the growth of the Internet, we have capabilities in the areas of Web and messaging application security as well as iden management. Addressing the fundamental issues of application security has requi is embodied in our Enterprise Gateway Security portfolio solutions listed below.

TrustedSource® - Global Intelligence to Protect Your Organization

TrustedSource technology is the most precise and comprehensive Internet host re world, and a cornerstone of our solutions. TrustedSource uses data collected from worldwide to assign a reputation "score" to each sender encountered. This score is Computing products to enable them to quickly and accurately reject unwanted traffi

Web Gateway Security Portfolio

Secure Computing offers a complete portfolio of Web Gateway Security appliances

Secure Computing offers a complete portfolio of Web Gateway Security appliances from malware, data leakage, and Internet misuse, and ensure policy enforcement, r a productive application environment. Through our Trusted Source technology we time literally millions of entities connected to the internet worldwide and provide up- behavior analysis to create a "reputation score" which can be used to determine wh Enterprise network should be allowed to occur. We also employ the most sophistic signature-based techniques for stopping Malware as well as patented content analy data leakage.

Web Gateway Security product solutions

Webwasher®
Webwasher® offers best-of-breed security solutions that p traffic from all types of Web-borne threats (malware, virus and outbound traffic from data leakage of proprietary infor

SmartFilter®
SmartFilter® enables organizations to understand and mc while taking effective steps to provide appropriate control access.

Messaging Gateway Security Portfolio

Secure Computing is the global market leader in Messaging Gateway Security. We portfolio of innovative, layered security solutions to stop inbound and outbound mes integrated, best-of-breed, and technologically superior appliance. Secure Computir maximum availability and unmatched security, effectiveness and global enterprise r multiple messaging protocols including email, instant messaging, and Webmail. Se leverages this capability using global intelligence with our TrustedSource technolog reputation system based upon real-time intelligence gathered from a worldwide net sensors. The combination of easy-to-manage appliances with sophisticated centrali clean, efficient communications, eliminating both inbound and outbound risks.

Messaging Gateway Security product solutions

IronMail®
IronMail® delivers a centrally managed, integrate messaging gateway security appliance for enterp sizes.

IronIM™
The IronIM™ instant messaging security applianc solution that integrates policy to secure, log, mon enterprise IM communications.

IronNet™
The IronNet™ appliance monitors all outbound in reviewed for compliance violations and subject to corporate policies regarding compliance violation

SecureEdge™
SecureEdge™ is a hardened appliance positione mail system, applying TrustedSource technology at the network border.

RADAR
RADAR™ was developed to protect an organizat
whether by detecting and stopping Phishing scan
fixing PCs.

TrustedSource
Secure Computing developed TrustedSource, the
comprehensive Internet host reputation system ir

Network Gateway Security Portfolio

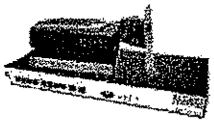
The definition of the "Enterprise Edge" has evolved significantly since the advent of
workforces, extranets, distributed applications, and an environment of highly sophis
has forced enterprises to deploy an array of security applications to provide service
IDS/IPS, anti-virus, anti-spam and more.

As a pioneer in firewall technology and unified threat management, Secure Comput
products carry the highest possible Common Criteria certification and have never o
eleven years across thousands of deployments. Our ability to leverage TrustedSou
real-time host profiling to our arsenal of security decision-making criteria makes Se
Gateway Security Portfolio truly unique.

Network Gateway Security product solutions

Sidewinder Network Gateway Security Appliance
Proven to be the most comprehensive security gatewa
the Sidewinder® Security Appliance consolidates all m
functions into a single system to defend against known

CyberGuard TSP
TSP appliances are designed to protect mid-sized to la
both known and zero-hour attacks, using a hybrid archi
stateful packet filtering, seven layer inspection, and sec
enforcement.

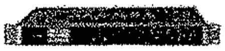


SnapGear Security Appliance
SnapGear™ is a complete office-in-a-box Internet secu
businesses, with wide area networking tools normally c
enterprise-class devices.

CommandCenter
CommandCenter™ is Secure Computing's enterprise-c
management solution that enables you to implement se
easily, and accurately across your entire security infras

Identity and Access Management

Establishing the identity of users accessing corporate network resources, remotely
well as enforcing the policies governing the scope of these interactions, are critical
Gateway Security. Accordingly, key components of our strategy are a dedicated Ide
Management appliance that combines strong authentication, remote access, and p
authentication form factors, including tokens.



SafeWord SecureWire
SafeWord® SecureWire™ is a powerful identity and acce
appliance that provides anywhere, anytime remote acces
data resource in the network for all remote and internal cc

SafeWord
By authenticating users through VPNs, Citrix, dial-up, and
SafeWord® provides trusted access to corporate applicati
anywhere. SafeWord tokens deliver single-use passcodes
vulnerabilities of passwords.

Security Services
Our award winning support team offers hands-on installation, training services, and
www.securecomputing.com/goto/support

Milestones

- **May 2007**, Secure Computing Releases Next Generation of SnapGear SME Appliance with TrustedSource
- **May 2007**, Secure Computing Launches Industry's First Self-Serve Domain
- **April 2007**, Secure Computing Awarded Three New Patents for Web and M Security technologies
- **March 2007**, SC Magazine Bestows 'Best Buy' Honors on Secure Computir
- **March 2007**, Secure Computing's Web Gateway Security Ranks #1 in Inde Security Study
- **February 2007**, Secure Computing's Sidewinder Wins SC Magazine Reade Enterprise Firewall
- **January 2007**, Sidewinder 7.0 and Webwasher 6.5 releases announced
- **January 2007**, Latest version of market-leading reputation system TrustedS
- **January 2007**, Secure Computing introduces Sidewinder 7.0
- **January 2007**, Webwasher 6.5 announced
- **December 2006**, Sidewinder G2 received Editor's Choice Award for functio Communications Week
- **December 2006**, Secure Computing honored with Best of 2006 awards for PremierAccess, and Webwasher
- **December 2006**, Secure Computing receives Reader Trust Finalist status fi Sidewinder G2, IronIM, and SmartFilter
- **December 2006**, Introduced the sleek, new carabiner-style Alpine token
- **November 2006**, Webwasher 6.0 released
- **October 2006**, Secure Computing positioned in the Leaders Quadrant for G Boundary 2006
- **October 2006**, Webwasher ranked the number 1 product for detecting the r Magazine
- **October 2006**, Secure Computing holds Messaging Security Conference in
- **September 2006**, SnapGear awarded CRN Test Center's "Recommended" Wireless Access for SMEs

- **September 2006**, Secure Computing announces SafeWord SecureWire 50
- **August 2006**, Secure Computing closes acquisition of CipherTrust
- **August 2006**, Sidewinder G2 named Best MidMarket Product of the Year b Magazine
- **July 2006**, CyberGuard TSP achieves Common Criteria Certification using
- **July 2006**, Secure Computing announces intention to merge with CipherTrı Security
- **June 2006**, Secure Computing named to FORTUNE Small Business Fastes Companies List
- **June 2006**, Secure Computing Positioned in Challengers Quadrant of Lead Quadrant
- **June 2006**, Secure Computing Named to Business 2.0's 100 Fastest-Growi Companies List
- **June 2006**, Announces Worldwide Channel Launch of Webwasher Secure Suite
- **May 2006**, IronMail named SC Magazine's Best Security Solution for Health
- **May 2006**, Announces SafeWord PremierAccess 4.0
- **May 2006**, Sidewinder G2 Security Appliance Cryptographic Module for Se 140-2 Validation
- **May 2006**, Secure Computing Honored with Reader Trust Awards from SC SafeWord, SmartFilter
- **April 2006**, Extends TSP Portfolio to Support Unified Threat Management
- **April 2006**, Launches SafeWord SecureWire Identity and Access Managem
- **March 2006**, Announces SnapGear Family of security appliances available worldwide channel
- **March 2006**, Secure Computing wins VARBusiness Magazine's 5-Star Rati
- **February 2006**, Information Security Magazine names IronMail 2006 Produ
- **February 2006**, Announced Zero-hour Attack Protections technology inside
- **January 2006**, Acquires CyberGuard Corporation
- **November 2005**, Secure Computing reaches 1,000 Partner Milestone
- **September 2005**, Announces relationship with McAfee's line of SCM applia technology
- **August 2005**, Celebrates Sidewinder G2 10-year flawless record of Zero cc
- **August 2005**, Releases SmartFilter 4.1
- **June 2005**, Launches Sidewinder G2 Security Reporter
- **May 2005**, Sidewinder G2 first to achieve Common Criteria Certification usi Application Firewall Medium Robustness Protection Profile
- **April 2005**, Secure Computing announces partnership with Sophos
- **March 2005**, Secure Computing wins VARBusiness Magazine's 5-Star Rati
- **February 2005**; SmartFilter wins SC Magazine Global Award
- **February 2005**, Announced 2005 new Sidewinder G2 lineup, including new
- **December 2004**, Sidewinder G2 named Product of the Year by Information
- **September 2004**, Secure Computing announces new version of SafeWord
- **September 2004**, Launched Safeword PremierAccess 3.2
- **September 2004**, Launched SafeWord RemoteAccess, Cisco compatible
- **August 2004**, Sidewinder G2 Security Appliance receives EAL4+ Certificati
- **June 2004**, Launched SmartFilter 4.0
- **January 2004**, Accelerated PartnersFirst Program by turning all but a selec channel partners
- **January 2004**, Announced Sidewinder G2 Security Appliance line
- **November 2003**, Announced SafeWord for Nortel Networks
- **October 2003**. Announced SafeWord for Check Point

- **October 2003**, Announced final acquisition of N2H2
- **August 2003**, SmartFilter receives OPSEC certification
- **July 2003**, Sidewinder G2 Firewall receives ICSA IPSec 1.1 certification
- **April 2003**, Sidewinder G2 Firewall achieves Common Criteria EAL4+ certil
- **April 2003**, Announced SafeWord for Citrix MetaFrame
- **February 2003**, Announced SmartFilter v3.2
- **January 2003**, Announced Sidewinder G2 Firewall and Sidewinder G2 Ente
- **January 2003**, SmartFilter available on Cisco Routers
- **January 2003**, John McNulty named CEO of the year for Network Security
- **June 2002**, Unveiled next generation firewall plans
- **May 2002**, Announced SafeWord PremierAccess v3.1
- **April 2002**, Gauntlet awarded Common Criteria EAL4 level certification
- **February 2002**, Announced acquisition of Gauntlet firewall and VPN busine
- **January 2002**, Announced Sidewinder Appliance
- **December 2001**, Announced SecureAlliance program
- **November 2001**, Sidewinder receives ICSA IPSec certification
- **October 2001**, Introduced SafeWord PremierAccess access control softwa authentication and authorization for e-Business applications
- **July 2001**, Sidewinder first firewall accepted into a Common Criteria Evalua
- **April 2001**, Introduced industry's first Embedded Firewall product
- **April 2001**, Announced $100,000 e-Security Challenge
- **1999**, Sidewinder receives Firewall of the Year from Network Magazine
- **1995-1996**, Acquired technology for two market leading access control prod authentication, and SmartFilter® URL filtering
- **1995**, Announced Sidewinder challenge
- **1995**, Introduced the world's first truly secure proxy-based firewall, Sidewin
- **1995**, Initial Public Offering
- **1989**, Spun off as Secure Computing Corporation
- **1984**, Secure Computing started as the Secure Computing Technology Cer

Secure Computing is a global leader in Enterprise Gateway Security software solutions. Powered by our Truste provides real-time web and messaging reputation scoring, our award winning portfolio of email, Web, and appli provide anti-spam, anti-virus, anti-phishing, anti-malware, and anti-spyware prevention and protection to help e Secure Computing's security software and network appliances also provide data leakage prevention, regulator auditing and reporting, strong authentication, and identity management.

Home | Contact | Privacy Policy

Enterprise Gateway Security | Internet Security | Internet Security Solutions | Network Security | Network security software | Network Manageme Software | Web Security | Web Gateway Security | Content Filtering | Web Filtering | Messaging Security | Messaging Gateway Security | Email ! Reputation Score | Reputation System | TrustedSource | Network Gateway Security | Firewall | Application Firewall | Security Appliance | VPN | I Criteria | Unified Threat Management | UTM Security | Identity Management | Access Control | Authentication | Strong Authentication | Radius Authentication | Password | Online Banking | Anti-malware | Anti-phishing | Anti-spam | Spam Block Spam Prevention | Anti-spyware | Anti-virus | Virus Blocker | Virus Signature | Virus Protection | Auditing & Reporting | Data I Regulations Compliance | CIPA Compliance

SECURE COMPUTING

+1.800.379.4944 Toll Free
+1.408.979.6572 International

Select Language

Search

Resources Products Try/Buy Support Partners Company

Locator

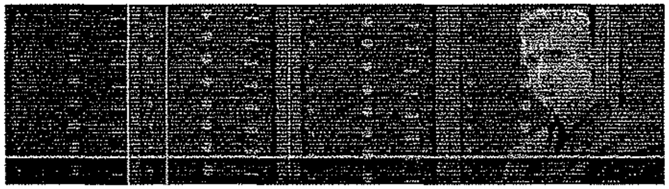
Printable Version

Network Gateway Security

- Features / Benefits
- Product Family / Models
- Resources/Papers/FAQ
- Technologies
- Certs/Testing/Compliance
- Awards/Recognition/News
- Support and Education
- How to Try/Buy
- SnapGear

Home > Network Gateway Security > Sidewinder >> Origin of Sidewinder

Web Gateway | Messaging Gateway | Network Gateway | Identity and Access



Next Steps

Sales Chat

Sidewinder

The Origin of Sidewinder® G2 Security Appliance

Security has always played a balancing act with usability. Any expert will acknowledge that it's simple to create a totally secure computer: you simply unplug every connection, including power, encase the thing in concrete, and *surround it with guards. By the same token, a pair of wire cutters provides the perfect network* firewall:·cut your Internet connections and we guarantee you won't suffer from Internet-based attacks.

On the other hand, computers aren't intended to be keepsakes. They're intended to be working devices that provide services and that share information with other computers. However, such sharing must be controlled, and typical computer software has always been unreliable. In the early 1970s, the U.S. Air Force organized "tiger teams" to evaluate the security of computers by trying to break in to them. At one of their first status meetings, a tiger team member handed a slip of paper to the programmer in charge of a computer's password security software - and the paper contained the programmer's own password! Computer security had improved by the late 1980s, but officials at the National Security Agency (NSA) still didn't consider the available computers to be secure enough for NSA's future needs. The NSA turned to Honeywell's Secure Computing Technology Center (SCTC) to build a highly secure computer to run the Unix operating system. The resulting system, called LOCK, tried to achieve the highest possible degree of security by using the most sophisticated software development techniques available. In 1989, Honeywell spun off SCTC into a private company, and that company became Secure Computing Corporation.

A major reason the government gave the job to Secure Computing was because of a key innovation developed by LOCK's chief engineer, W. Earl Boebert, and his colleagues. That innovation, *Type Enforcement® technology*, was a mechanism that would provide strong security and also make it much easier to analyze and verify the system's security. Type enforcement technology essentially divided the computer into separate compartments, like those used to make an oceangoing ship less likely to sink even if part of the hull is breached. Inside the computer, type enforcement prevented a misbehaving program in one part of the system from being able to damage other parts of the system, even if the program managed to acquire administrative privileges.

While LOCK provided security features important to the NSA's high security applications, it didn't appeal to commercial customers. LOCK relied on "high end" microcomputers, which were much larger and more expensive than the Intel-based PCs that had rapidly multiplied during the early 1990s. At the same time, developers like Marcus Ranum at Digital Equipment Corporation were experimenting with devices called "Internet firewalls" to protect sites from remote attacks across the Internet.

Boebert took a look at the fundamentals of the LOCK design and the techniques developed on LOCK to protect network traffic, and realized they could yield an exceptionally strong firewall. Starting in early 1994, Boebert assembled a small team of developers to develop a commercially viable version of the LOCK system to play host to a firewall. They decided to do this by modifying BSDI UNIX to incorporate type enforcement, and to host the resulting system on Intel hardware. This effort created the SecureOS® that formed the basis for the original Sidewinder, and is still used today in the Sidewinder G2 Firewall.

By fall of 1994, the prototype system was ready. Boebert and his team had a lot of confidence in the system, but they needed to put it to the test. Boebert was an old hand at computer security and was familiar with the hacker underground. He knew that hackers were usually young people motivated by curiosity and that they often established reputations by cracking tough systems. With the Sidewinder prototype attached to the Internet, Boebert made visits to hacker bulletin boards and discussion groups. He posted the Sidewinder's Internet address along with notes describing it as an interesting target. Although this produced a few weak attempts to hack the machine, Boebert was impatient to see some serious attempts.

To turn up the heat, Boebert went public with the Sidewinder prototype, announcing its presence on Internet security discussion groups as well as hacker bulletin boards. Boebert taunted his audience, declaring that this new Sidewinder device was obviously too much for hackers to handle. He also provided a few technical details to whet people's appetites.

This was not the first such challenge in the history of computer security, but it may have been the first really successful one. A few years earlier, a vendor introduced a secure computer at the National Computer

Security Conference and challenged attendees to try to break in. An attendee broke in within minutes of the challenge's announcement.

Following Boebert's announcement, hundreds hackers visited the Sidewinder site, tried their classic exploits, but never broke through the type enforcement barrier. A few declared success after trying a few simple exploits, not realizing that they were like prisoners who had broken out of cells, only to remain trapped inside the cellblock. More sophisticated hackers tried to manipulate programs running in neighboring domains and achieved little further success.

In October 1994, Secure Computing officially announced Sidewinder at the National Press Club, and followed up with demonstrations at the National Computer Security Conference. Secure Computing also officially announced the Sidewinder Challenge, which invited anyone on the Internet to try to break the type enforcement protections of Sidewinder. The winner was offered bragging rights and a custom-made leather jacket with the Sidewinder snake logo on the back.

The Sidewinder Challenge easily weathered the conference without falling to a successful attack, and the challenge site remained on the Internet for years, absorbing thousands of attacks. Starting in 1995, it also made several visits to the DefCon hacker conventions to server as a special target for the attendees (at one convention, they gave up on Sidewinder and instead found success hacking the web site for the movie "Hackers"). The Challenge remained unbroken. Over the years, the leather jacket was augmented with cash awards. These awards reached $100,000 before the Challenge was retired at the Black Hat Briefings in 2001.

The Sidewinder Challenge has often been copied but never been equaled. Vendors occasionally post a challenge to attack their product, but no other product has withstood the years of attempts racked up by Sidewinder without hacker successes.

Experts in computer security will readily point out that security challenges don't really prove that a product is safe, since you can't be sure that every possible attack has been applied to the product. Still, the Sidewinder Challenge and what it shows about the product's strength and robustness have impressed many customers, including government experts looking for the strongest security products.

To continue the chronicle of history, in February 2002, Secure Computing acquired the Gauntlet firewall business from Network Associates. The goals of the acquisition were to transition the many thousands of marquis Gauntlet customers around the world to the Secure Computing firewall support programs and then over time to upgrade them to a new unified firewall platform. The new next generation firewall platform was promised to be built as a combination of the key features from both Gauntlet and Sidewinder utilizing a best-of-breed approach.

In January of 2003, Secure released their new, next-generation firewall: the newly branded Sidewinder G2 Firewall. Sidewinder G2 began shipping in early Q1 2003 as Secure's flagship firewall offering. Also in January 2003, Secure released the Sidewinder G2 Enterprise Manager™, a new security appliance that delivers both single-point policy management and a central audit-log & configuration back-up repository for hundreds of distributed Sidewinder G2 firewalls. The Sidewinder G2 Firewall now provides performance capabilities equal to or exceeding other firewall vendors, while still providing unmatched security features and advantages.

In January 2004, the Sidewinder G2 Firewall became the all-new Sidewinder G2 Security Appliance - which has gone on to win numerous industry and analyst awards.

Sidewinder G2® Security Reporter™ was launched in June, 2005 - which is a powerful, easty-to-use, and cost-effective security event analysis and reporting solution for the Sidewinder G2 Security Appliance.

In August of 2005, the Sidewinder G2 Security Appliance celebrated it's 10-year anniversary with a flawless record of **zero compromises**!

This classic **Screen Saver** accompanied the initial release of Sidewinder in 1994.

Today, the Sidewinder G2 Security Appliance ***still*** remains the world's strongest firewall and VPN gateway.

Home | Contact | Privacy Policy | Disclaimer | Sitemap

Enterprise Gateway Security | Internet Security | Internet Security Solutions | Network Security | Network security software | Network Management | Security Policy | Security Software | Web Security | Web Gateway Security | Content Filtering | Web Filtering | Messaging Security | Messaging Gateway Security | Email Security | Global Intelligence | Reputation Score | Reputation System | TrustedSource | Network Gateway Security | Firewall | Application Firewall | Security Appliance | VPN | Intrusion Detection | Common Criteria | Unified Threat Management | UTM Security | Identity Management | Access Control | Authentication | Strong Authentication | Radius Authentication | Password | Online Banking | Anti-malware | Anti-phishing | Anti-spam | Spam Blocker | Spam Filter | Spam Prevention | Anti-spyware | Anti-virus | Virus Blocker | Virus Signature | Virus Protection | Auditing & Reporting | Data Leakage | Regulations Compliance | CIPA Compliance

US005864683A

United States Patent [19]

Boebert et al.

[11] **Patent Number: 5,864,683**

[45] **Date of Patent: Jan. 26, 1999**

[54] **SYSTEM FOR PROVIDING SECURE INTERNETWORK BY CONNECTING TYPE ENFORCING SECURE COMPUTERS TO EXTERNAL NETWORK FOR LIMITING ACCESS TO DATA BASED ON USER AND PROCESS ACCESS RIGHTS**

[75] Inventors: **William E. Boebert**, Minneapolis; **Clyde O. Rogers**, White Bear Lake; **Glenn Andreas**, Fridley; **Scott W. Hammond**, Maplewood; **Mark P. Gooderum**, St. Louis Park, all of Minn.

[73] Assignee: **Secure Computing Corporartion**, Roseville, Minn.

[21] Appl. No.: **322,078**

[22] Filed: **Oct. 12, 1994**

[51] **Int. Cl.**^{6} **G06F 15/173**

[52] **U.S. Cl.** **395/200.79**; 395/187.01; 380/4; 380/25

[58] **Field of Search** 380/9, 25, 49, 380/23, 21, 24, 4; 395/187.01, 491, 600, 200.16, 608, 800, 200.79; 370/85.3

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,956,615	5/1976	Anderson et al.	380/24
4,104,721	8/1978	Markstein et al.	364/200
4,177,510	12/1979	Appell et al.	364/200
4,442,484	4/1984	Childs, Jr. et al.	364/200
4,584,639	4/1986	Hardy	395/650
4,621,321	11/1986	Boebert et al.	395/608
4,648,031	3/1987	Jenner et al.	364/200
4,701,840	10/1987	Boebert et al.	395/800
4,713,753	12/1987	Boebert et al.	380/4
4,870,571	9/1989	Frink	395/200.16
4,885,789	12/1989	Burger et al.	380/25
4,888,801	12/1989	Foster et al.	380/21
4,914,568	4/1990	Kodosky et al.	364/200
4,914,590	4/1990	Loatman et al.	364/419
5,093,914	3/1992	Coplien et al.	395/700
5,124,984	6/1992	Engel	370/94.1

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

0554182A1	4/1993	European Pat. Off.	H04L 29/06
2 287 619	9/1995	United Kingdom .	
96/35994	11/1996	WIPO .	

OTHER PUBLICATIONS

J.A. Adam, "Meta-matrices," *IEEE Spectrum*, 26 (Oct. 1992).

J.A. Adam, "Playing on the Net," *IEEE Spectrum*, 29 (Oct. 1992).

N.J. Belkin et al., "Information Filtering and Information Retrieval: Two Sides of the Same Coin?", *Commun. of the ACM*, 35, 29 (1992).

T.F. Bowen et al., "The Datacycle Architecture," *Commun. of the ACM*, 35, 71 (1992).

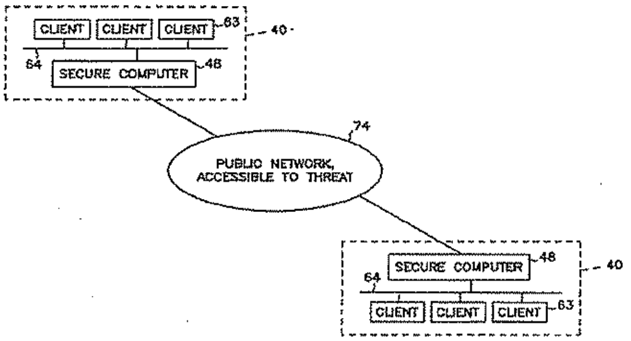
(List continued on next page.)

Primary Examiner—Thomas C. Lee
Assistant Examiner—David Ton
Attorney, Agent, or Firm—Schwegman, Lundberg, Woessner & Kluth, P.A.

[57] **ABSTRACT**

A system and method for the secure transfer of data between a workstation connected to a private network and a remote computer connected to an unsecured network. A secure computer is inserted into the private network to serve as the gateway to the unsecured network and a client subsystem is added to the workstation in order to control the transfer of data from the workstation to the secure computer. The secure computer includes a private network interface connected to the private network, an unsecured network interface connected to the unsecured network, wherein the unsecured network interface includes means for encrypting data to be transferred from the first workstation to the remote computer, a server function for transferring data between the private network interface and the unsecured network interface and a filter function for filtering data transferred between the remote computer and the workstation.

21 Claims, 14 Drawing Sheets



U.S. PATENT DOCUMENTS

5,153,918	10/1992	Tuai	380/25
5,204,961	4/1993	Barlow	395/725
5,228,083	7/1993	Lozowick et al.	380/9
5,263,147	11/1993	Francisco et al.	395/491
5,272,754	12/1993	Boerbert	380/25
5,276,735	1/1994	Boebert et al.	380/21
5,303,303	4/1994	White	380/49
5,305,385	4/1994	Schanning et al.	380/49
5,311,593	5/1994	Carmi	380/23
5,329,623	7/1994	Smith et al.	395/187.01
5,333,266	7/1994	Boaz et al.	395/200
5,355,474	10/1994	Thuraisngham et al.	395/600
5,414,833	5/1995	Hershey et al.	395/187.01
5,416,842	5/1995	Aziz	380/30
5,418,951	5/1995	Damashek	395/600
5,455,828	10/1995	Zisapel	370/85.3
5,485,460	1/1996	Schrier et al.	370/94.1
5,511,122	4/1996	Atkinson	380/25
5,548,646	8/1996	Aziz et al.	380/23
5,550,984	8/1996	Gelb	395/200.17
5,566,170	10/1996	Bakke et al.	370/60
5,583,940	12/1996	Vidrascu et al.	380/49
5,586,260	12/1996	Hu	395/200.2
5,604,490	2/1997	Blakley, III et al.	340/825.31
5,606,668	2/1997	Shwed	395/200.11
5,615,340	3/1997	Dai et al.	395/200.17
5,619,648	4/1997	Canale et al.	395/200.01
5,623,601	4/1997	Vu	395/187.01
5,636,371	6/1997	Yu	395/500
5,644,571	7/1997	Seaman	370/401
5,673,322	9/1997	Pepe et al.	380/49
5,684,951	11/1997	Goldman et al.	395/188.01
5,689,566	11/1997	Nguyen	380/25

OTHER PUBLICATIONS

P.W. Foltz et al., "Personalized Information Delivery: An Analysis of Information Filtering Methods," *Commun of the ACM*, 35, 51 (1992).

D. Goldberg et al., "Using Collaborative Filtering to Weave an Information Tapestry," *Commun. of the ACM*, 35, 61 (1992).

S.T. Kent, "Internet Privacy Enhanced Mail," *Commun. of the ACM*, 36, 48 (1993).

K. Lee et al., "A Framework for Controlling Cooperative Agents," *Computer*, 8 (Jul. 1993).

S. Loeb, "Architecting Personalized Delivery of Multimedia Information," *Commun. of the ACM*, 35, 39 (1992).

K. Obraczka et al., "Internet Resource Discovery Services," *Computer*, 8 (Sep. 1993).

L. Press, "The Net: Progress and Opportunity," *Commun. of the ACM*, 35, 21 (1992).

M.F. Schwartz, "Internet Resource Discovery at the University of Colorado," *Computer*, 25 (Sep. 1993).

Commun. of the ACM, 35, 28 (Dec. 1992).

Copy of PCT Search Report dated Apr. 9, 1996 by Areste Canosa for Application No. PCT/US95/12681 (8 pages).

S.M. Bellovin et al, entitled Network Firewalls, *IEEE Communications Magazine*, 32, No. 9, pp. 50–57, dated Sep. 1994.

J. Bryan, entitled Firewalls For Sale, *BYTE*, pp. 99–100, 102, 104–105, dated Apr. 1995.

F.T. Grampp, entitled UNIX Operating System Security, *AT&T Bell Laboratories Technical Journal*, 63, No. 8, pp. 1649–1672, dated Oct. 1984.

Lee Badger, et al., "Practical Domain and Type Enforcement for UNIX", *1995 IEEE Symposium on Security and Privacy*, pp. 66–77, (May, 1995).

William R. Bevier, et al., "Connection Policies and Controlled Interference", *The Eighth IEEE Computer Security Foundations Workshop*, IEEE Computer Society Technical Committee on Security and Privacy, pp. 167–176, (Jun. 1995).

B. B. Dillaway, et al., "A Practical Design For A Multilevel Secure Database Management System", *American Institute of Aeronautics and Astronautics, Inc.*, pp. 44–57, (Dec. 1986).

Todd Fine, et al., "Assuring Distributed Trusted Mach", *IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 206–218, (1993).

J. Thomas Haigh, et al., "Extending the Noninterference Version of MLS for SAT", *IEEE Transactions on Software Engineering*, vol. SE–13, No. 2, pp. 141–150, (Feb, 1987).

Yuet C. Lee, et al., "Multimedia: Full Service Impact on Business, Education, and the Home", *SPIE –The International Society for Optical Engineering*, vol. 2617, pp. 143–150, (Oct. 1995).

Richard E. Smith, "Sidewinder: Defense in Depth Using Type Enforcement", *International Journal of Network Management*, pp. 219–229, (Jul.–Aug. 1995).

"100% of Hackers Failed to Break Into One Internet Site Protected by Sidewinder", News Release, Secure Computing Corporation (Feb. 16, 1995).

"Internet Security System Given 'Product of the Year' Award", News Release, Secure Computing Corporation (Mar. 28, 1995).

"SATAN No Threat to Sidewinder™", News Release, Secure Computing Corporation (Apr. 26, 1995).

Ancilotti, P., et al., "Language Features for Access Control", *IEEE Transactions on Software Engineering*, SE–9, 16–25 (Jan. 1983).

Damashek, M., "Gauging Similarity with n–Grams: Language–Independent Categorization of Text", *Science*, 267, 843–848, (10 Feb. 1995).

Lampson, B.W. "Dynamic Protection Structures", *AFIPS Conference Proceedings, vol. 35, 1969 Fall Joint Computer Conference*, Las Vegas, NV, 27–38 (Nov. 18–20, 1969).

Schroeder, M.D., et al., "A Hardware Architecture for Implementing Protection Rings", *Communications of the ACM*, 15, 157–170 (Mar. 1972).

Thomsen, D., "Type Enforcement: the new security model", *SPIE Proceedings*, vol. 2617 from Oct. 23–24, 1995, 143–150, (1995).

Warrier, U.S., et al., "A Platform for Heterogeneous Interconnection Network Management", *IEEE J. on Selected Areas in Communications*, vol. 8, No. 1, 119–126, (1990).

Wolfe, A, "Honeywell Builds Hardware for Computer Security", *Electronics*, 14–15 (Sep. 2, 1985).

"Answers to Frequently Asked Questions About Network Security", Secure Computing Corporation, pp. 1–41 & pp. 1–16, (Sep. 25, 1994).

"Sidewinder Internals", Product Information, Secure Computing Corporation,. 16 p., (Oct. 12, 1994).

"Special Report: Secure Computing Corporation and Network Security", *Computer Select*, 13 p., (Dec. 1995).

McCarthy, S.P., "Hey Hackers! Secure Computing Says You Can't Break into this Telnet site", *Computer Select*, 2 p., (Dec. 1995).

Merenbloom, P., "Network 'Fire Walls' Safeguard LAN Data from Outside Intrusion", *Infoworld*, LAN Talk, p. 69 & add'l page, (Jul. 25, 1994).

Metzger, P., et al., "IP Authentication using Keyed MD5", Network Working Group, Request for Comment No. 1828, 6 p., (Aug. 1995).

Smith, R.E., "Constructing a High Assurance Mail Guard", Secure Computing Corporation (Appeared in the Proceedings of the National Computer Security Conference), 7 p., (Oct. 1994).

Stadnyk, I., et al., "Modeling User's Interests in Information Filters", *Communications of the ACM*, 35, 49–50, (Dec. 1992).

Stempel, S., "IpAccess — An Internet Service Access System for Firewall Installations", *IEEE*, 31–41, (1995).

Stevens, C., "Automating the Creation of Information Filters", *Communications of the ACM*, 35, 48, (Dec. 1992).

White, L.J., "A Firewall Concept for Both Control–Flow and Data–Flow in a Regression Integration Testing", *IEEE*, 262–271, (1992).

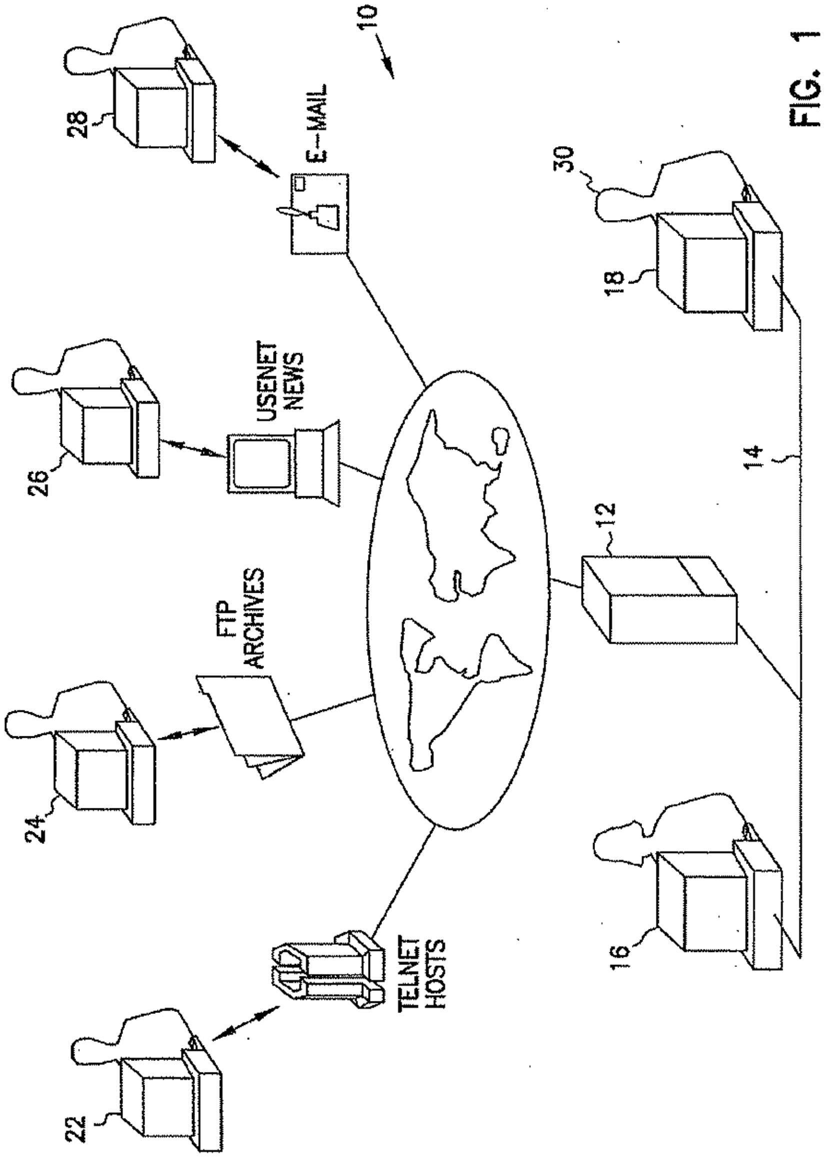

10
28
E-MAIL
26
USENET NEWS
24
FTP ARCHIVES
22
TELNET HOSTS
30
18
14
12
16

FIG. 1

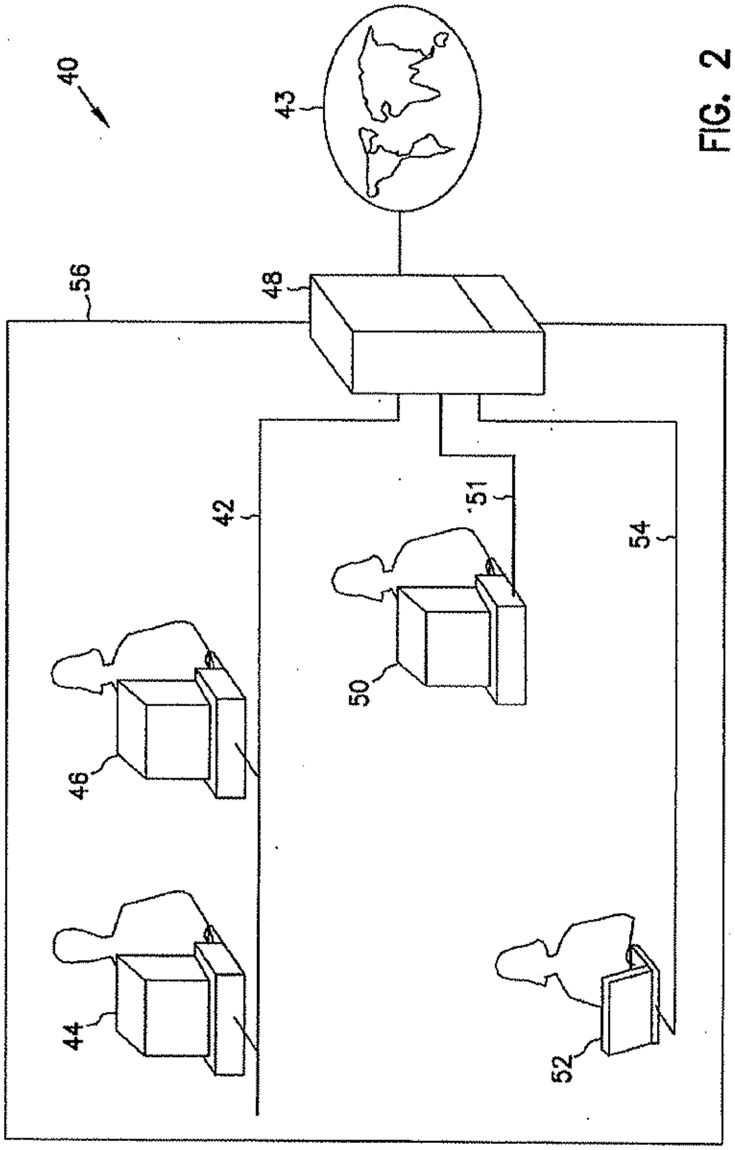


FIG. 2

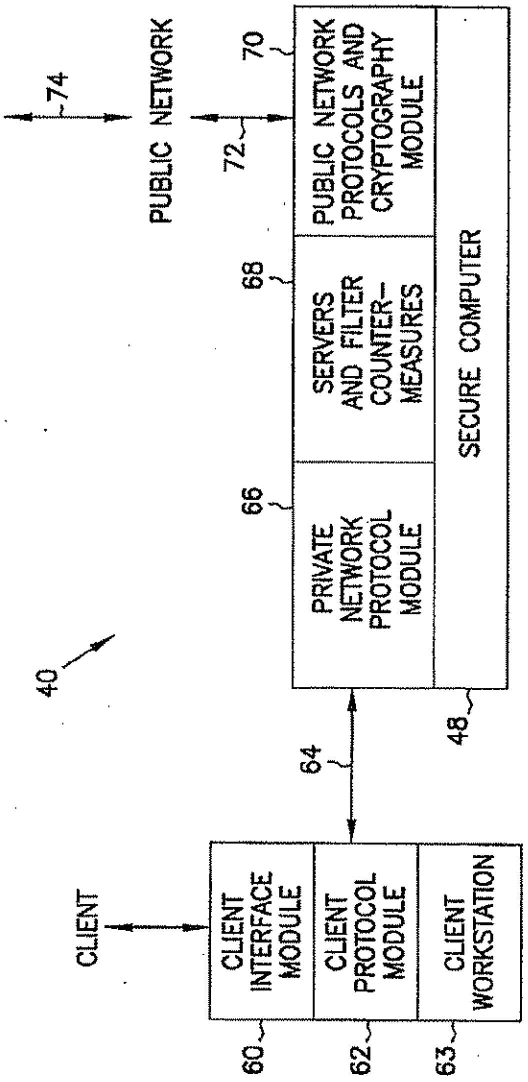

40
CLIENT
CLIENT INTERFACE MODULE
60
CLIENT PROTOCOL MODULE
62
CLIENT WORKSTATION
63
64
PRIVATE NETWORK PROTOCOL MODULE
66
SERVERS AND FILTER COUNTER-MEASURES
68
PUBLIC NETWORK PROTOCOLS AND CRYPTOGRAPHY MODULE
70
SECURE COMPUTER
48
72
PUBLIC NETWORK
74

FIG. 3

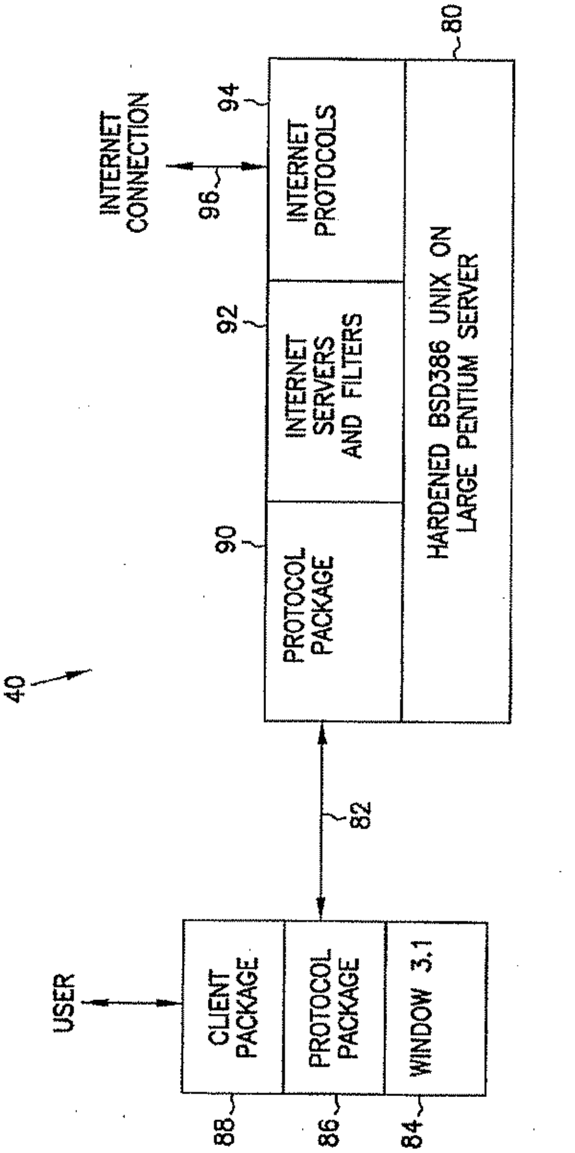

40
USER
CLIENT PACKAGE
PROTOCOL PACKAGE
WINDOW 3.1
88
86
84
82
PROTOCOL PACKAGE
INTERNET SERVERS AND FILTERS
INTERNET PROTOCOLS
HARDENED BSD386 UNIX ON LARGE PENTIUM SERVER
90
92
94
96
80
INTERNET CONNECTION

FIG. 4

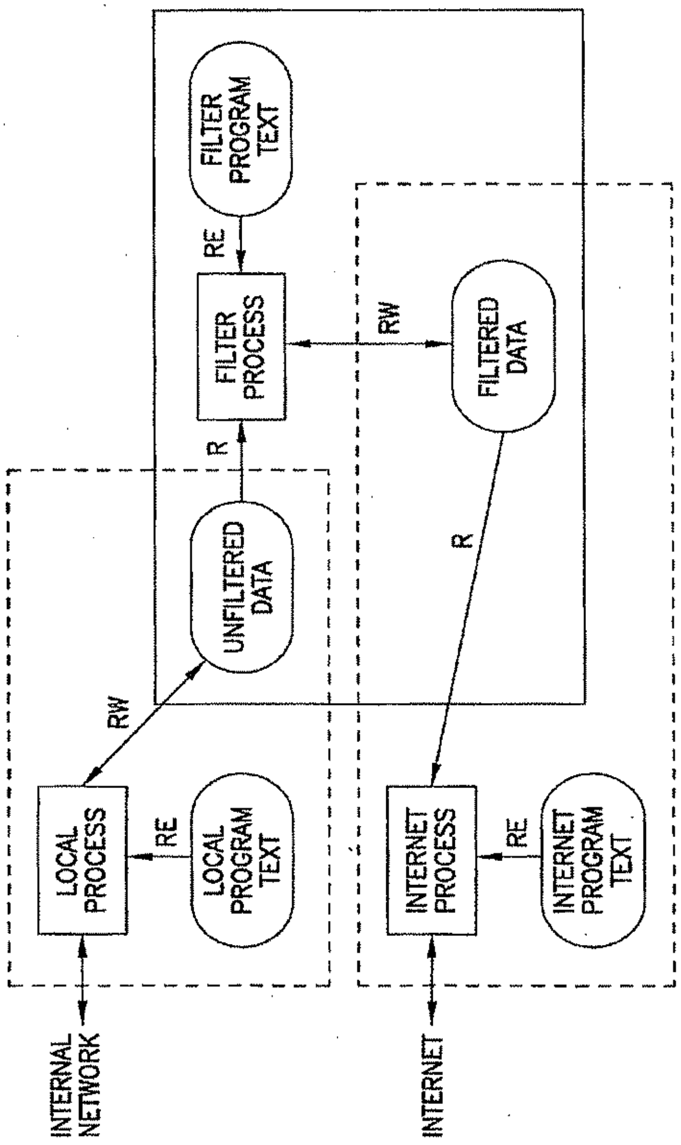

FILTER PROGRAM TEXT
RE
FILTER PROCESS
RW
FILTERED DATA
R
UNFILTERED DATA
R
RW
LOCAL PROCESS
RE
LOCAL PROGRAM TEXT
INTERNET PROCESS
RE
INTERNET PROGRAM TEXT
INTERNAL NETWORK
INTERNET

FIG. 5A

TYPE →

DOMAIN ↓	UNFIL DATA	FIL DATA	CLIENT PGM	FLTR PGM	NET PGM
LOCAL	RW	▨	RE	▨	▨
FILTER	R	RW	▨	RE	▨
INTERNET	▨	R	▨	▨	RE

R: READ ONLY
RW: READ AND WRITE
RE: READ AND EXECUTE
▨: NO ACCESS ALLOWED

FIG. 5B

Subype	Source Code Name	Usage
file	file	Files that are private to the creating Domain.
directory	diry	Directories; not checked.
socket	sock	Reserved.
fifo	fifo	Fifos; not checked.
device	devi	Reserved.
port	port	Reserved.
executable	exec	Executable; effective Domain is not changed.
gate	gate	Executable; effective Domain is set to creator field of full Type name for duration of execution.

FIG. 6

Attribute Name	Meaning
ddt_read	Process may read (fetch) data from object.
ddt_write	Process may modify object.
ddt_rename	Process may rename object.
ddt_exec	Process may execute contents of object. Will only by assigned to subtypes gate and exec and will never be combined with ddt_write.
ddt_trigger	Trigger an alarm signal to Rover monitoring facilities as a side effect of granting access.
ddt_chcreator	If effective Domain of the process = creator field of Type, then process can change creator field.
ddt_destroy	Process may destroy the object.

FIG. 7

Syscall	Type of Executable Argument	Old Real Domain	Old Effective Domain	New Real Domain	New Effective Domain	Remarks
fork	No Argument	Mail	Mail	Mail	Mail	Child Process Spawned
execve	Mail:exec	Mail	Mail	Mail	Mail	New Executable
execve	$Sys:exec	Mail	Mail	Mail	Mail	New Executable
makedomain (with MIME as domain name)	Mail:exec	Mail	Any	MIME	MIME	New Executable, as with execve
makedomain (with MIME as domain name	SMTP:gate	Mail	Any	MIME	SMTP	New Executable, as with execve
changedomain (with MIME as domain name)	No Argument	Mail	Any	MIME	MIME	Continue With Same Executable
execve	SMTP-gate	Mail	Any	Mail	SMTP	Implicit Gating, New Executable
gate (with SMTP as domain name)	No Argument	Mail	Any	Mail	SMTP	Explicit Gating, Continue With Same Executable
ungate	No Argument	Mail	SMTP	Mail	Mail	Explicit Exit from Gate

FIG. 8

Privilege Name[a]	Meaning
can_ch_type	Can execute the ch_type or fch_type syscalls.
suppress_su_alarm	Can exercise super-user privilege without tripping alarm.
admin_reboot	Can transition to admin ("Greendome") mode.
can_set_clock	Can set or adjust the system clock.
can_setlogin	Can set user login name.
is_startup	Can perform functions required for startup only.

[a] Note that there is no "can_ch_domain" privilege: the ability to change either the real or effective Domain of a process is controlled by the DIT.

FIG. 9

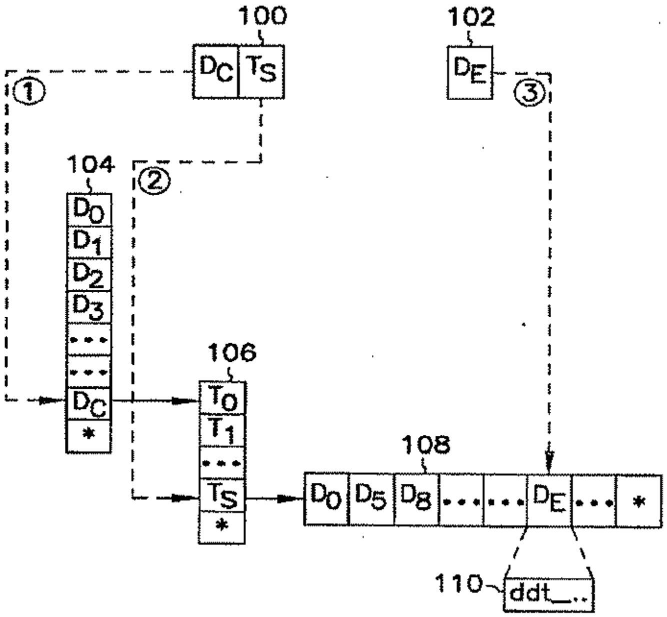

100
D_C
T_S
102
D_E
①
②
③
104
D_0
D_1
D_2
D_3
• • •
• • •
D_C
*
106
T_0
T_1
• • •
T_S
*
108
D_0
D_5
D_8
• • •
• • •
D_E
• • •
*
110
ddt_..

FIG. 10

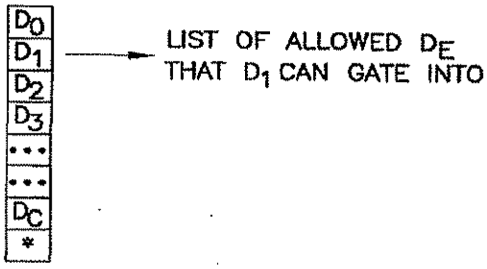


FIG. 11

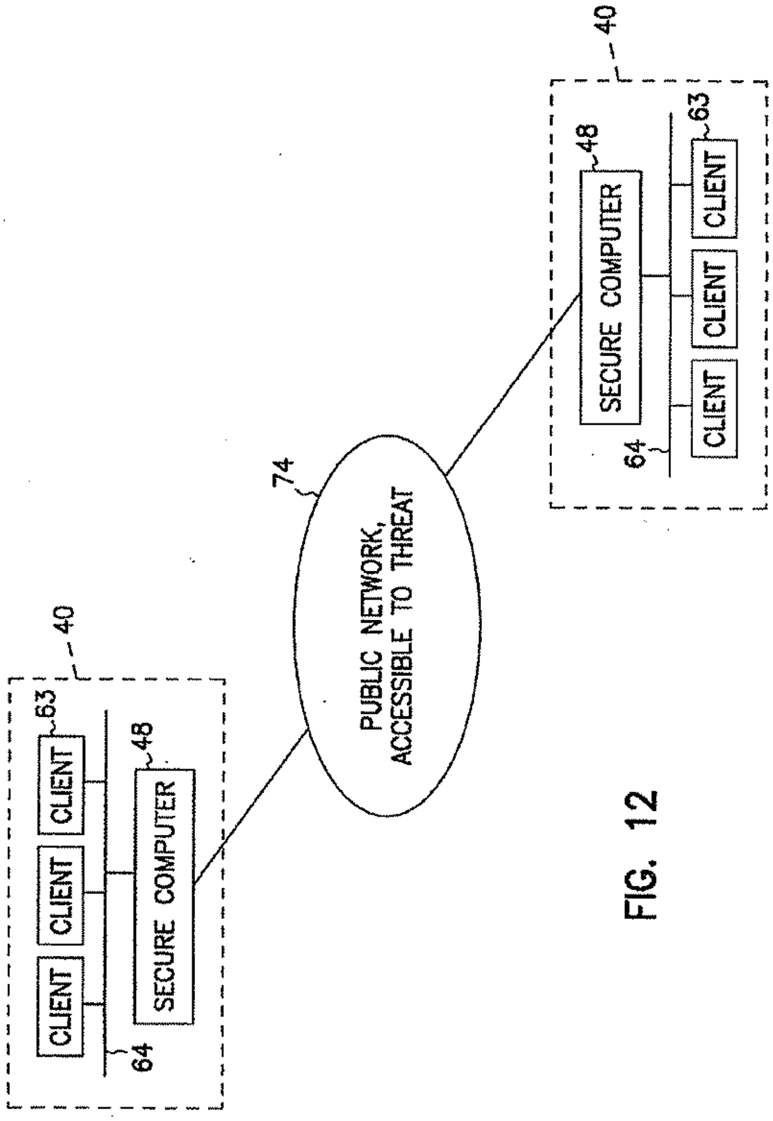

40
48
SECURE COMPUTER
63
CLIENT
CLIENT
CLIENT
64
74
PUBLIC NETWORK,
ACCESSIBLE TO THREAT
40
63
CLIENT
CLIENT
CLIENT
48
SECURE COMPUTER
64

FIG. 12

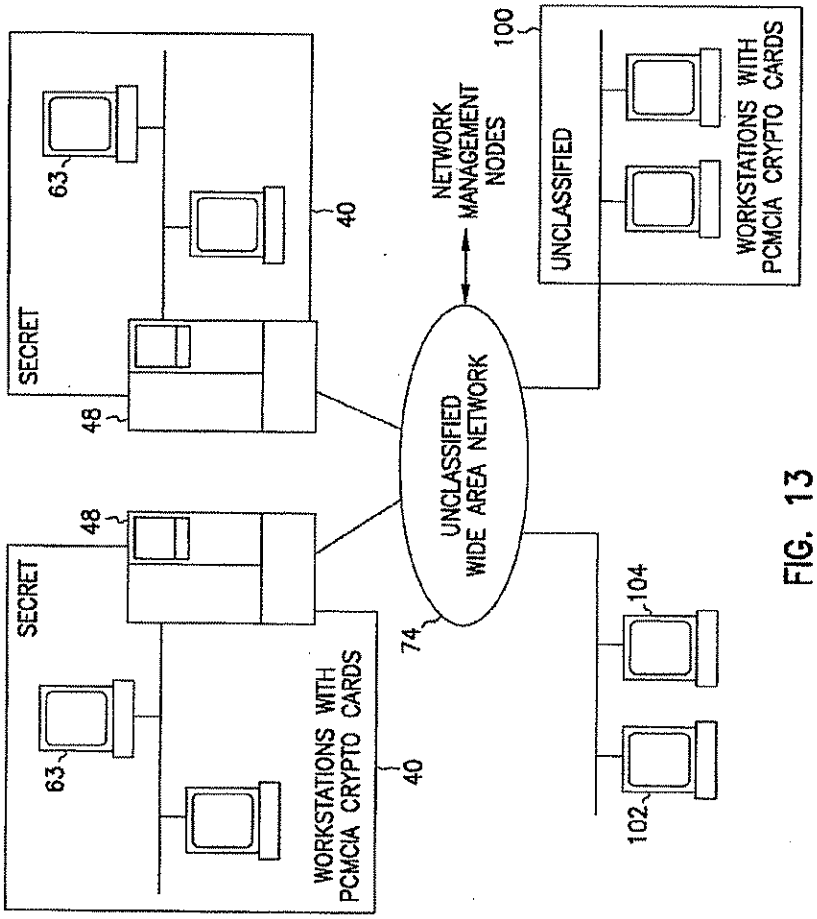

SECRET
63
40
48
SECRET
48
63
WORKSTATIONS WITH PCMCIA CRYPTO CARDS
40
74
UNCLASSIFIED WIDE AREA NETWORK
NETWORK MANAGEMENT NODES
100
UNCLASSIFIED
WORKSTATIONS WITH PCMCIA CRYPTO CARDS
104
102

FIG. 13

SYSTEM FOR PROVIDING SECURE INTERNETWORK BY CONNECTING TYPE ENFORCING SECURE COMPUTERS TO EXTERNAL NETWORK FOR LIMITING ACCESS TO DATA BASED ON USER AND PROCESS ACCESS RIGHTS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to computer security, and more particularly, to an apparatus and method for providing secure access to a wide-area network.

2. Background Information

Advances in computer and communications technology have increased the free flow of information within networked computer systems. While a boon to many, such a free flow of information can be disastrous to those systems which process sensitive or classified information. In a typical networked computer system, one or more workstations are connected over a network to a host computer or server. These workstations may range from low-cost personal computers to powerful UNIX processors. In such a system the workstations, servers and even the connecting networks may all be at great risk of a security breach.

In developing a strategy for reducing the potential and consequences of a security breach (i.e. a computer security policy), one must assume that competent and dedicated individuals will mount active attacks on the computer system's security mechanisms. These individuals are called the threat. The threat seeks to find vulnerabilities which can be exploited to cause a part of the computing system to operate in violation of its owner's security policy. Threats fall into two broad classes: Insiders and Outsiders.

Insiders are those individuals who have been granted some level of legitimate privilege and then abuse that privilege. An example of an insider in the noncomputer world is a bookkeeper who uses his or her legitimate access to account records to embezzle. An example in the computer world is a systems administrator who uses his or her legitimate access to a computer system to generate fraudulent billings, payable to a corporation owned by the administrator. Concern for insider actions also extends to individuals who, through ignorance, incompetence or improper direction, cause security policy to be violated intentionally.

Outsiders are those individuals who have no legitimate privilege on the system but who can exploit vulnerabilities to gain access to it. An example of an outsider in the noncomputer world is a burglar, who exploits weaknesses in locks and alarms to steal from a safe or lockbox. An example of an outsider in the network world is the "hacker" who takes control of a networked computer away from its legitimate owners.

The risk of security breach is compounded when a pathway is provided from the internal, private network to an external wide-area network such as the Internet. Internet is a loose conglomeration of networks connected by a standard network protocol. The lure of access to Internet is the vast amounts of information that can be accessed by the user; the danger is that there are little or no controls on what individuals have access to and what they may do with that access. Therefore, access to Internet can provide an open door for exploitation of your own network by a variety of threats.

In effect, a wide-area network such as Internet serves as a threat multiplier. Networks such as Internet have evolved as fora for the free exchange of ideas. This fact can be exploited by threats seeking to access or subvert a private network. For instance, the global connectivity of such a

network means that data taken from a private network can be moved around the world very quickly. To compound this problem, Internet contains a number of very large data archives which can be used to store data transferred or posted from private networks. Hackers have also used the global connectivity of wide-area networks such as Internet to directly manipulate computer facilities on the internal network (by such mechanisms as trying unlikely combinations of requests or commands) or to inject malicious software into the machine. Malicious software, which is able to do the threat's bidding remotely and without direct control, can be injected manually or by such technical mechanisms as "viruses" or "worms." (One such self-replicating piece of malicious software was responsible for a well publicized attack on computers connected to the Internet a few years ago.)

Internet protocols that have been developed to-date were not designed for security. For instance, Usenet news can be used by ignorant or disgruntled employees to post company proprietary information in publicly accessible space. In some cases, this posting can be done anonymously (e.g. by using an anonymous file transfer mode or by posting the data to an anonymous server). In addition, the proprietary nature of data may be obscured by encrypting the data via one of a number of free, easily accessible cryptographic packages.

In addition, since the standard Unix password is reusable, it is subject to capture and abuse by outsider threats. For instance, the use of reusable passwords means that each password is vulnerable to being "sniffed out" and captured. Once captured the password can be used by an inside or an outside threat to gain access to a site. In addition, if the password belongs to someone with administrative privilege, the threat can use the captured password to gain administrative privileges on the internal network. The threat can then use that privilege to install a permanent "trapdoor" in order to ensure future access.

This combination of features makes the Internet particularly vulnerable to attack. A potential buyer of stolen information can anonymously post a solicitation along with his public key; potential sellers can then encipher the information desired with that public key and post it, secure in the knowledge that only the solicitor will be able to decipher it.

The existence of an active threat places requirements on a private network which are significantly different from the superficially similar problem of providing reliable service. A reliability engineer can take advantage of the low probability of certain phenomenon, and choose not to respond to them because they are so unlikely. A security engineer cannot do this; a vulnerability, however obscure and unlikely, will be actively sought out by the threat, publicized to persons of like mind, and exploited over and over once discovered. Countermeasures must therefore be developed which effectively close, or prevent the exploitation of, each system vulnerability.

A number of countermeasures have been proposed to reduce the vulnerability of networked systems. These countermeasures share three characteristics:

1) It takes a secret to keep a secret. All information security mechanisms are based on the use of secrets which are shared by authorized individuals an kept from unauthorized ones. The secrets may be transformed, compressed or hidden inside protected hardware, but in every security architecture there is one set of values, which, if known, would lead to the compromise of the whole system.

2) Vulnerabilities always exist. It is no more possible to achieve perfect security than it is to achieve perfect reliability; in fact, it is much less possible because you must assume that the threat is actively working to discover the system vulnerabilities.

3) Threats escalate continuously. Installation of a given set of countermeasures does not eliminate the threat; it simple spurs it on to greater efforts to find ways of circumventing them.

These three common factors then pose the following problems for the countermeasures engineer:

1) Protecting the secrets that keep the secrets. This is highest priority requirement, for loss of these values would lead to catastrophic breaches of security.
2) Making vulnerabilities hard to find. The embodiment of the security mechanisms must be such that it is difficult for the threat to obtain details of their operation, or instances of them on which experiments may be performed.

The countermeasures proposed to date have focussed on either preventing the transfer of data or on encrypting the data using known cryptographic methods in order to render it more difficult to compromise.

One method proposed for the prevention of unauthorized exploitation of the private network by inside or outside threats is an Internet "firewall". "Firewalls" implement a security policy based on the routing information contained in individual packets transferred to and from the wide-area network. They look only at the headers of the packets and then make decisions based on where the packet is going and where it came from. Typically, "firewalls" direct packets to a dedicated application machine which has a limited configuration of software. This application machine is then connected to a second router that limits its access to a specific set of internal systems.

A typical Internet "firewall" system **10** is shown in FIG. 1. In FIG. 1, system **10** includes a router **12** connected over an internal network **14** to workstations **16** and **18**. Router **12** is also connected to a wide-area network **20** such as Internet. Router 12 runs Internet "firewall" software intended to inspect packet based traffic and remove or reroute packets meeting a predefined criteria.

"Firewalls" are header sensitive, not content sensitive. Therefore they are subject to various forms of attack. For instance, a hacker **22** may construct a packet having a header which looks like a header passed by the firewall. Such a packet will slip unnoticed past router **10** and onto one or more workstations **16, 18**. In addition, a threat **24** may be able to access sensitive data on network **14** through the file transfer protocol ("FTP"). As noted above, a buyer **26** of stolen data may use Usenet news to solicit transfer of proprietary data from venal or disgruntled employees. Finally, a threat **28** may work in conjunction with a subverted employee **30** to transfer proprietary information via encrypted electronic mail or anonymous FTP.

Therefore, the Internet firewall approach has the following disadvantages:

1) This approach is vulnerable to attacks which construct fake header information (such as that by hacker **22** above). The theory of such attacks is well known; it is only a matter of time before turnkey scripts for mounting them become globally available on Internet.
2) A "firewall" is an "all-or-nothing" approach to security. If an attacker gets through the "Firewall", then the internal network on the other side lies naked and unprotected against effectively undetectable trojan horse attacks.
3) "Firewalls" can be difficult to configure correctly and even more difficult to keep secure because they have to be reconfigured as you modify your internal network.
4) "Firewalls" cannot make security decisions based on data content, because they only see the data after it has been cut into packets and rearranged in the course of transmission.

5) "Firewalls" limit, in arbitrary and irrational ways, the user's ability to interact with the Internet.
6) "Firewalls" require special "proxy" software for many Internet services. This means that there is a slow and costly development step required to "secure" a new service using the "Firewall" technique.
7) "Firewalls" require extra hardware and network connections, which increases cost and administrative overhead.

The cryptographic countermeasures proposed to date have focussed on encrypting the data using known cryptographic methods in order to render it more difficult to compromise. Cryptography operates by performing mathematical transforms on data so that it is rendered unintelligible to an outside observer. In order for the data to be retrieved, the transform is based on a second set of values called keying material. It is the keying material that is, in this case, the secret that keeps the secrets. Since both the writer and the authorized reader of the data must have equivalent keying material, the central problem in cryptography is key management: the safe and reliable delivery of equivalent keying material to both ends of the writer-reader axis.

Cryptographic transforms use mathematical algorithms of great complexity and sophistication. In order to provide real-world security it is also necessary, however, that the embodiment or implementation of the algorithm be not only correct but also free of vulnerabilities or side effects which can be exploited by the threat.

One commonly used class of cryptographic algorithms is called secret-key or symmetric. Such algorithms are called symmetric because the same element or value of keying material is used both to encipher (scramble) and to decipher (unscramble). They are called secret-key because that keying material must be kept secret at both the writer and the reader ends of a communication. Secret-key systems require a some degree of prearrangement between the writer and the reader, so that the identical values of keying material are in place in advance of communication. As such, secret-key cryptography is most suited for communication amongst a closed community, where membership in the community is known a priori. Simple changes in key distribution patterns can be used to add or delete individuals from the community.

Another class of cryptographic algorithms is called public-key or asymmetric. Such algorithms are called asymmetric because two mathematically related elements of keying material are required: a public key, which is used to encipher but which cannot be used to decipher (unscramble), and a private key, which is the only value that can decipher. The corresponding private key, which is the secret that keeps the secret, is closely held. The public key, since it cannot be used to decipher, can be widely disseminated. By this means a secret message can be sent without explicit prearrangement: the writer obtains the reader's public key from some service akin to a telephone directory, enciphers the message, and sends it with the knowledge that only the reader holds the private key that can decipher it.

A form of public-key algorithm can also be used to authenticate, or sign, data. In this operation the private key is used to compute a value which is mathematically related to the data, called a digital signature. The private key is used so that only the holder of that private key can establish the distinctive value of the signature. The mathematics of the operation are such that the corresponding public can be used to determine the validity of the signature. Thus only one person can sign, but any individual with access to the public key service can check the signature.

Public-key cryptography is most suited for communication within an open community, where it is desired to have secret and/or authenticated communication without prior arrangement. Adding individuals to the community is relatively simple, but deleting individuals is difficult.

Cryptography has the following uses in information security:

1) Protection of communications links where the transmissions can be easily intercepted.

2) Protection of electronic mail where the messages may be forwarded through sites not under the control of the writer or the authorized reader of the message.

3) Protection of data stored on removable media or media which is exposed to the possibility of physical theft.

4) Authentication, where the knowledge of a shared secret is used to verify the identity of an individual or a machine.

The most sophisticated approaches to protecting data transferred the unsecured Internet network are through the application of Global Cryptography the Client workstation, so that data is enciphered at the source and deciphered/destination. The principal application of this approach is to electronic mail. Cryptography can be implemented in software, as in the Privacy Enhanced system, or in personal tokens which combine the cryptographic mechanisms individual's certificate, as in the MOSAIC program.

A less sophisticated approach is to apply the cryptography wide-area network. Historically, there have been two ways to do Encryption and End-to-End Encryption.

In the Link Encryption approach, all bits coming node and onto the network are enciphered. This requires that the have an identical cryptographic device and compatible keying source. The disadvantage of link encryption is that all bits those used to route packets over a packet-switched network a packet-switched network from working.

To permit the use of cryptography over packet-switched networks, the technique of End-to-End Encryption was devised. In this technique, only the packet contents are encrypted, and the critical routing information is left as plaintext. The "ends" in End-to-End encryption are typically multi-user servers and not individual workstations, so that the problem of getting compatible keying material at each end is reduced to manageable proportions.

Neither data encryption nor the use of Internet "firewalls" address the array of vulnerabilities inherent to connection of an internal, private network to an external, wide-area network such as the Internet. What is needed is a comprehensive and integrated security policy and apparatus for preventing exploitation of private network resources by both internal and external threats.

SUMMARY OF THE INVENTION

The present invention provides a secure wide-area access system comprising a secure computer, an internal network and a workstation connected across the internal network interface, a public network interface, public network program code used to communicate through the public network interface to a public network, private network program code used to communicate through the internal network interface to the workstation and security policy program code for enforcing a Type Enforcement security mechanism to restrict access of a process to data.

According to another aspect of the present invention, a method of protecting a computer system connected to an unsecured external network is described. The method comprises the steps of providing a secure computer, wherein the secure computer comprises security policy program code for enforcing a Type Enforcement security mechanism to restrict access of a process to data, connecting the Type Enforcement based secure computer to the private network and establishing an assured pipeline for the transfer of data and programs between the private network and the external network through the secure computer. The step of establishing an assured pipeline includes the steps of placing processes within domains, wherein the step of placing processes within domains includes the step of assigning processes received from the external network to an external domain, assigning types to files and restricting access by processes within the external domain to certain file types.

According to yet another aspect of the present invention, a secure server is described for use in controlling access to data stored within an internal network. The secure server comprises an administrative kernel and an operational kernel, wherein the operational kernel includes security policy program code for enforcing a Type Enforcement security mechanism to restrict access of a process received from the external network to data stored on the internal network and wherein the administrative kernel is restricted to execution only while isolated from the internal network.

According to yet another aspect of the present invention, the secure server comprises a processor, an internal network interface, connected to the processor, for communicating on an internal network and an external network interface, connected to the processor, for communicating to an external network. The processor includes server program code for transferring data between the internal and external network interfaces and security policy program code for enforcing a Type Enforcement security mechanism to restrict access of a process received from the external network to data stored on the internal network.

According to yet another aspect of the present invention, a system and method for the secure transfer of data between a workstation connected to a private network and a remote computer connected to an unsecured network. A secure computer is inserted into the private network to serve as the gateway to the unsecured network and a client subsystem is added to the workstation in order to control the transfer of data from the workstation to the secure computer. The secure computer includes a private network interface connected to the private network, an unsecured network interface connected to the unsecured network, wherein the unsecured network interface includes means for encrypting data to be transferred from the first workstation to the remote computer and a server function for transferring data between the private network interface and the unsecured network interface.

According to yet another aspect of the present invention, a system is described for secure internetwork communication across an unsecured network. First and second secure computers are connected to first and second private networks, respectively, and to each other across the unsecured network. The first and second secure computers include a private network interface and an unsecured network interface for secure transfer of data from the first secure computer to the second secure computer over the unsecured network. The unsecured network interface includes means for encrypting data to be transferred from the first secure computer to the second secure computer. A client subsystem is added to workstations connected to each private network in order to control the transfer of data from the workstation to the respective secure computer.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a representation of a router-based "firewall";

FIG. 2 is a system level block diagram representation of a secure wide-area access system according to the present invention;

FIG. 3 is a more detailed block diagram representation of one embodiment of the networked computer system of FIG. 2;

FIG. 4 shows one embodiment of the system of FIG. 3;

FIGS. *5a* and *5b* show the Type Enforcement mechanism used to prevent access, modification and/or execution of data objects without permission in a system such as that shown in FIG. 3;

FIG. 6 is a table of source code names of subtypes;

FIG. 7 is a table of access attributes;

FIG. 8 is a table of the new and effective Domains which result from particular syscalls;

FIG. 9 is a table listing the privileges which may be granted to a Domain;

FIG. 10 is a representation of steps taken in determining access privileges from the DDT;

FIG. 11 is a representation of steps taken in determining from the DIT the Domains a process can change to;

FIG. 12 is a system level block diagram of a wide area network connecting two organizational enclaves according to the present invention; and

FIG. 13 is a system level block diagram of another embodiment of a wide area network connecting two organizational enclaves according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the following Detailed Description of the Preferred Embodiments, reference is made to the accompanying Drawings which form a part hereof, and in which are shown by way of illustration specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

A secure wide-area access system 40 is shown in FIG. 2. In FIG. 2, an internal network 42 connects workstations 44 and 46 to secure computer 48. Internal network 42 is separated from a wide-area network 43 (such as Internet) by secure computer 48. Secure computer 48 is also connected to a system administrator workstation 50 through a dedicated line 51 and to a workstation 52 through a serial interface 54. Secure computer 48 and workstations 44, 46, 50 and 52 make up an organizational enclave 56 of data. The enclave is a "logical" enclave in that there is no requirement that the protected users and data be physically co-located, although such use of physical security measures is certainly possible.

It is important to isolate network 42 from network 43. To do this, secure computer 48 enforces an organizational security policy at the interface between internal network 42 and wide-area network 43. It must do so in the face of active threat from both insiders and outsiders, whether by direct manipulation, the insertion of malicious software, or a combination of both. The system must protect its clients against attacks from wide-area network 43, limit the damage done by subverted or incompetent clients, and be able to securely interact with clients of other systems 40 connected to wide-area network 43. It does this by surrounding the user with a set of protections that form organizational enclave 56.

Organizational enclave 56 consists of two main elements: a Client subsystem which operates on workstations 44, 46 and 52 and a set of servers and filters which operate on secure computer 48. In one embodiment, internal network 42 connecting each workstation 44 or 46 to secure computer 48 is protected and authenticated by Local Cryptography; Global Cryptography is used for protection and authentication on the wide-area network 43. In one embodiment, the Global Cryptography function uses local authentication plus endorsement or notarization by Secure Internetwork Services System to identify the source of a communication.

FIG. 3 illustrates one embodiment of the secure wide-area access system 40 shown in FIG. 2. In FIG. 3, a workstation 64 (e.g. workstation 44, 46 or 52) connected to secure

computer 48 over Private Network 63 (e.g. internal network 42 or serial interface 54) contains program code for communicating with secure computer 48 and through secure computer 48 to computers connected to wide-area network 43. Private Network 64 can be any means of communication, wired or wireless, which allows a workstation 63 to transfer data between the workstation and secure computer 48. In the example shown in FIG. 2, two embodiments of private network 64 are shown (internal network 42 and serial interface 54). It should be apparent that other embodiments of Private Network 64 can be implemented and the resulting system 40 would still fall within the scope of the present invention.

In one embodiment, the program code in workstation 63 includes a Client Interface Module 60 and a Client Protocol Module 62. Client Interface Module 60 accepts commands from, and displays results to, the user or Client. It can be embodied in a Graphical User Interface (GUI), a command line interface, or some combination of the two. Typical commands would be to prepare an electronic message, examine incoming messages, request files from other sites, or any other operations typical of computer network usage.

Client Protocol Module 62 implements the protocol used to communicate between workstation 63 and the secure computer 48. Client Protocol Module 62 can be implemented in either software or hardware, or a combination of both. In one embodiment, a Local Cryptography function integrated into Protocol Module 62 has the specialized task of protecting and authenticating traffic on internal network 42 only. Different protocols and different cryptographic methods can be used for different Clients, depending on Client preferences and such factors as the nature of the physical connection (dialup, Local Area Network, etc.) between the Client Workstation and the Secure Computer. It is most likely, though not required, that the closed nature of an organizational Client community (i.e. organizational enclave 56) will favor the use of secret-key cryptography in this module. In one embodiment, the Local Cryptography function is implemented in software in order to take advantage of software's flexibility and interoperability advantages over hardware.

In another embodiment, the Local Cryptography function is implemented as a module separate from but operating in conjunction with Client Protocol Module 62.

In secure wide-area access system 40 of FIG. 3, program code running on secure computer 48 is used to communicate through Private Network 64 to Client Protocol Module 62. In the embodiment shown in FIG. 3, the program code used to communicate with Client Protocol Module 62 is part of Private Network Protocol Module 66. In such an embodiment, Module 66 runs on secure computer 48 and interacts with Client Protocol Module 62 to provide protected and authenticated communication with workstation 63.

Likewise, program code running on secure computer 48 is used to communicate through a Public Network interface 72 to Public Network 74 (e.g. Internet). In the embodiment shown in FIG. 3, the program code used to communicate with Public Network 74 is part of Public Network Protocols and Cryptography Module 70. In such an embodiment, Module 70 runs on secure computer 48 and is used to provide protected and authenticated communication with individuals, sites, and other secure wide-area access systems 40 on Public Network 74. Different protocols and cryptographic methods may be used when communicating with different entities on Public Network 74. It is most likely, though not required, that the open nature of Public Network 74 will favor the use of public-key cryptography in this module.

Finally, program code running on secure computer **48** is used to implement servers and filter functions on secure computer **48**. In the embodiment shown in FIG. 3, the program code used to implement the server and filter functions is part of Servers and Filters Countermeasures 68. As such, the servers and filter countermeasures operate on the secure computer **48**. They provide user services, such as the delivery of electronic mail or the transfer of data files and also enforce the organizational security policy by filtering the transfer of information and intercepting disallowed contents, labels, and/or addresses.

Cryptography in Secure Systems

The principal requirement for secure use of cryptography is a safe and reliable method for distribution of keying material. Reliability is as important as safety because if the material is not available then the users of the system are faced with the unpleasant choice of either not using the cryptography (and thereby exposing their data to compromise or modification) or not transmitting. The key management requirements for a secret key system revolve around prearranged distribution of shared secrets. The key management requirements of public key systems revolve around insuring that the writer of a document to be enciphered obtains the public key which corresponds to the reader's private key. Since the consequences of obtaining the wrong public key can be a breach of security, public keys are digitally signed by a notary or local authority who attests to their validity. Such signed public keys, with other optional information about the holder of the corresponding private key, are called certificates.

Any effective key management system, and by extension any effective use of cryptography in a computer network, must also have facilities to solve the following problems:

1) Revocation. It must be possible to "take back" keying material so that an individual who was once authorized can have that authorization revoked.

2) Emergency rekey. It must be possible to "revive" the authorization of an individual if the keying material that grants the authorization is lost or destroyed.

3) Travelling user. The keying material that grants authorization to an individual must move around the network as the individual changes location.

Theoretically, the security of a cryptographic mechanism should rest only on the secrecy of critical keying material (all of it in a secret-key system, just the private part in a public-key system). As a practical matter, it is necessary to maintain protection of the mechanism for cryptography. This is especially true when the cryptographic device is partially or fully controlled by a computer system which may have been subverted through the use of malicious software. Such malicious software could cause the cryptographic device to be bypassed either physically, by routing sensitive data around it, or logically, by causing a coherent pattern to be imposed on the timing or other characteristics of the output. This is not a cryptographic problem per se, but rather one that arises in the systems context of cryptography combined with potentially vulnerable computers.

In the secure wide-area access system **40** of FIGS. **2** and **3**, the burden of maintaining protection of the mechanism of cryptography is placed on secure computer **48**. Secure computer **48** can be any type of machine whose features and/or implementation permits the operation of security-relevant functions to be trusted. Trusted computing systems have been proposed for limiting access to classified information to those who have a sufficient level of clearance. Such systems depend on identifying the user, authenticating (through password, biometrics, etc.) the user's identity and limiting that user's access to files to those files over which he or she has access rights. Such systems are described in U.S. Pat. Nos. 4,621,321; 4,713,753; and 4,701,840 granted to Boebert et al. and assigned to the present assignee, the entire disclosures of which are hereby incorporated by reference.

Typically, secure computers such as secure computer **48** provide safeguards through specialized hardware and software from direct attack on program code running in the secure computer. They have been developed to meet the following two objectives:

1) Limiting the privilege of users in a shared, or multiuser computer installation, so that malicious users cannot cause damage or compromise, and the effect of user error is minimized; and

2) Preventing damage or compromise that could result from the execution of malicious or erroneous software.

There have been two approaches to achieve the latter objective: exclusion, which seeks to prevent malicious software from entering the machine, and confinement, which allows the software into the machine and seeks to limit its effects. Existing secure computers fall into three broad classes:

1) Multilevel Secure Computers, which apply a confinement policy modelled on the U.S. Department of Defense system of data classification and personnel clearances. A Multi-Level Secure (MLS) Computer is capable of recognizing data of varying sensitivity and users of varying authorizations and ensuring that users gain access to only that data to which they are authorized. For example, an MLS computer can recognize the difference between company proprietary and public data. It can also distinguish between users who are company employees and those who are customers. The MLS computer can therefore be used to ensure that company proprietary data is available only to users who are company employees.

2) Type Enforcing Secure Computers, which apply a confinement policy based on data flows through software subsystems in the machine.

3) Special Purpose Secure Computers, which apply an exclusion policy to insure that no malicious software is inserted in them, and then perform special-purpose security-related functions.

Secure wide-area access system **40** of FIGS. **2** and **3** can make use of any of these classes of machines, although it is most suited to being implemented on a Type Enforcing Secure Computer.

A freestanding Secure Computer has the following preconditions for secure use:

1) Protection of mechanism: the security mechanisms, especially those embodied in software, must be protected from tampering or unauthorized modification. Since software mechanisms are prone to frequent update and improvement, there is a requirement for trusted distribution, that is, a means whereby administrators can be confident that the software they are installing is correct and proper.

2) User authentication: the security mechanisms often decide whether or not to allow an action based on the individual on whose behalf the action is being taken. There must be a method whereby the identity of a user can be authenticated.

In the case of a freestanding Secure Computer, physical controls are typically sufficient to protect mechanism and simple methods such as passwords are sufficient to authenticate user identities. Designers of secure computers assume that unauthorized individuals will use a variety of means, such as malicious code and active and passive wiretaps, to circumvent its controls. Trusted subsystems of a secure computer must therefore be designed to withstand malicious software executing on the untrusted subsystem, to confine the actions of malicious software and render it harmless. For instance, trusted computer systems based on host computers such as a Multilevel Secure (MLS) Computer make security breaches at the host computer more difficult by partitioning the system to isolate security critical (trusted) subsystems from nonsecurity critical (untrusted) subsystems. In a similar manner, in Type Enforcing (TE) Secure Computers executables residing within the secure computer can only be executed if the person requesting execution has execution privileges for that executable object. A further level of security can be achieved by preventing execution of any executable objects that have not been expressly recognized as a trusted executable by a trusted executable or by a system administrator.

In one embodiment of a TE-based system **40**, only trusted executables are permitted to execute within secure computer **48**. In such an embodiment, executables must first be reviewed and validated by a system administrator before they will be granted execution privileges on secure computer **48**

Secure computers do little, however, to prevent security breaches on the private network or at the workstation. One mechanism for avoiding such a breach is to authenticate the client to the secure computer over the network. The Local Cryptography function described above performs such a client authentication function. Another mechanism for avoiding a network-related breach is to invoke a trusted path, a secure communications path between the user and the trusted subsystem. A properly designed trusted path ensures that information viewed or sent to the trusted subsystem is not copied or modified along the way. A trusted path authenticates not only the client to secure computer **48** (as in Local Cryptography above) but also authenticates secure computer **48** to the client. As such, the trusted path mechanism guarantees that a communication path established between the trusted subsystem on secure computer **48** and the user cannot be emulated or listened to by malicious hardware or software.

Extension of the trusted path through the network to the user is, however, difficult. As is described in a previously filed, commonly owned U.S. patent application entitled "Secure Computer Interface" (U.S. Pat. No. 5,272,754 issued Dec. 21, 1993 to William E. Boebert), "active" and "passive" network attacks can be used to breach network security. Active attacks are those in which masquerading "imposter" hardware or software is inserted into the network communications link. For example, hardware might be inserted that emulates a user with extensive access privileges in order to access sensitive information. "Passive" network attacks include those in which a device listens to data on the link, copies that data and sends it to another user. The '754 patent describes a system and method for ensuring secure data communications over an unsecured network. That disclosure is hereby incorporated by reference. Operation of a trusted path in conjunction with an organizational enclave is described in U.S. Pat. No. 5,276,735, issued Jan. 4, 1994 to Boebert et al., the description of which is hereby incorporated by reference.

In one embodiment, therefore, communication between Client Protocol Module **62** and Private Network Protocol Module **66** is made secure through the establishment of a Trusted Path between workstation **63** and secure computer **48** for all critical transfers.

Security Policy within the Secure Wide-Area Access System

The term security policy has acquired two meanings in the art:

1) The statement used by organizations and individuals to describe the objectives of their security activity, and to assign roles and responsibilities.

2) The rules used by a Secure Computer to determine whether or not certain actions may be performed.

In the latter case there are two kinds of policies:

2a) Label-based, in which the decisions are made on the basis of tag, or internal label, which is associated with a data object such as a file. The contents of the file are not examined by the decision-making mechanism.

2b) Content-based, in which the decisions are made on the basis of the contents of the file, message, or other data object.

Secure computers are required to perform the following tasks:

1) Protect data while it is being processed in unencrypted form. Certain operations, such as computations, editing, and transformation from one electronic message format to another can only be performed on data in unencrypted or cleartext form. Operations in encrypted, or ciphertext form, are generally limited to storage and transmission.

2) Enforce content-based security policies. Since such enforcement requires examination of contents, those contents must be in intelligible plaintext form.

3) Enforce individual roles and control the exercise of privilege. Cryptography inherently provides a binary or "all or nothing" privilege mechanism: either one possesses a decryption key, in which case one can read the data and then do whatever one pleases with it, or one does not possess the decryption key and operations on the data are prevented.

In a computer network, cryptography requires the following services from a Secure Computer:

1) Reliable and safe key management and distribution, including enforcement of limited roles for privileged individuals.

2) Protection of cryptographic mechanism from abuse by malicious software.

Correspondingly, Secure Computers require the following services from cryptography:

1) Authentication of user identities.

2) Protection of software mechanisms through trusted distribution.

3) Protection of data during storage or transmission in exposed environments such as a Public Network.

Underlying Principles of the Secure Wide-Area Access System

The first principle of system **40** is that the security services and alarms are centralized in a protected facility (secure computer **48**) which is under the administrative control of a limited number of authorized individuals. Secure computer **48** can, and in general will, be physically protected to prevent unauthorized tampering or modification. In this way a greater degree of trust can be placed in its operation that in the operation of Client workstations **63**, which are exposed, and in some cases portable. Centralization means that security alarms are signalled only to authorized administrators who have privileges on secure computer **48**; this facilitates response to insider attacks. Centralization also means that new services and countermeasures can be implemented simply by changing program code or hardware on secure computer **48**; such changes will be immediately available to, and imposed upon, all Clients on Private Network **64**.

Secure wide-area access system **40** distinguishes between local authentication and protection, which takes place within the more protected confines of a Private Network **64**, and global authentication and protection, which takes place over a Public Network **74** shared with potentially hostile parties. All information is decrypted and examined in plaintext form by Filter Countermeasures **68** on secure computer **48**. This permits the imposition of content-based organizational security policies and detailed audit of Client interactions with

Public Network 74. It also permits the intelligent transformation of data from one format to another when crossing the boundary between the Private Network 64 and Public Network 74. This ability is especially important in the case of electronic mail, where a large number of incompatible formats are in place.

A Type Enforcing Secure Wide-Area Access System

One embodiment of secure wide-area access system **40** of FIG. **3** is illustrated in the block diagram of FIG. **4**. In FIG. **4**, system **40** includes a secure computer **80** connected across a private network **82** to one or more workstations **84**. Workstations **84** are Intel-based IBM compatible personal computers running Windows 3.1 on the Microsoft DOS operating system. Protocol package **86** implements the protocol used to communicate between workstation **84** and secure computer **80**. Network **82** may be implemented, for instance, as a serial or a network connection. In one embodiment, network **82** uses a TCP/IP protocol. In such an embodiment, protocol package **86** is a software package used to establish a WINSOCKET to network **82** on workstation **84**. In one such embodiment, a Local Cryptography function is integrated into protocol package **88** in order to protect and authenticate traffic on network **82**.

Client package **88** accepts commands from, and displays results to, the user or Client. It can be embodied in a Graphical User Interface (GUI), a command line interface, or some combination of the two. Typical commands would be to prepare an electronic message, examine incoming messages, request files from other sites, or any other operations typical of computer network usage.

In secure wide-area access system **40** of FIG. **4**, program code running on secure computer **80** is used to communicate through Private Network **82** to protocol package **86**. In one embodiment, secure computer **80** is an Intel Pentium-based machine running a hardened form of BSD386 Unix. A system based on a 90 MHz pentium with 32 megabytes of memory, 2 gigabytes of hard disk space, a DAT tape for backup and a CD-ROM for software loads has been found to be adequate.

In the embodiment shown in FIG. **4**, the program code used to communicate with protocol package **86** is part of protocol package **90**. In such an embodiment, package **90** runs on secure computer **80** and interacts with protocol package **86** to provide protected and authenticated communication with workstation **84**. For instance, a Local Cryptography function may consist of software which executes on workstation **84** to establish client authentication at login. In such a system, when a user logs into network **82**, a message is sent from workstation **84** to secure computer **80**. Secure computer **80** responds with a number (in one embodiment, this is a seven digit number) which is sent unencrypted to protocol package **86** on workstation **84**. Protocol package **86** then generates a request, through client package **88**, to the user to enter his or her personal identification number (PIN). Protocol package **86** takes the PIN and combines it with a predefined number stored on workstation **84** to form a DES encryption key. That DES encryption key is then used to encrypt the number received from secure computer. The encrypted number is sent to secure computer **80**, where it is decrypted. If the correct machine number and PIN number were used for that particular user, secure computer **80** will be able to reconstruct exactly the number it sent to workstation **84**. If not, an error is generated and an entry is made in the audit log. In one embodiment, active spoofing countermeasures are then executed in an attempt to keep the threat in the vicinity of workstation **84**.

Once the client is authenticated, communication on network **82** is in clear text.

Likewise, program code running on secure computer **80** is used to communicate through a Public Network interface to a Public Network. In the example shown in FIG. **4**, the

public network is Internet. In such an embodiment, the program code used to communicate with the Internet is part of an Internet protocols **94** which communicates with computers on the Internet through Internet connection **96**. Internet protocols **94** runs on secure computer **80** and is used to provide protected and authenticated communication with individuals, sites, and other secure wide-area access systems **40** over the Internet. Different protocols and cryptographic methods may be used when communicating with different entities on the Internet. In one embodiment, a tcp wrapper package operating in Internet protocols **94** is used to sit on the external, public network so that information about external probes can be logged. It is most likely that the open nature of Public Network 74 will favor the use of public-key cryptography in this module.

Finally, program code running on secure computer **80** is used to implement servers and filter functions on secure computer **80**. In the example shown in FIG. **4**, the program code used to implement the server and filter functions is part of Internet Servers and Filters **92**. As such, the servers and filter countermeasures operate on secure computer **80**. They provide user services, such as the delivery of electronic mail or the transfer of data files and also enforce the organizational security policy by filtering the transfer of information and intercepting disallowed contents, labels, and/or addresses.

As noted above, in one embodiment secure computer **80** is an Intel Pentium-based machine running a hardened form of Berkeley's BSD386 Unix. In that embodiment, BSD386 is hardened by adding a Type Enforcement mechanism which restricts the access of processes to data. Type Enforcement operates in conjunction with page access control bits in the virtual page translator of the Pentium to control access to objects stored in secure computer **80** memory. To accomplish this, system calls in the basic BSD386 kernel were modified as shown later in this document so that Type Enforcement checks cannot be avoided. Certain other system calls were either disabled or had certain options disabled.

In the hardened BSD386 according to the present invention, Type Enforcement controls are enforced by the kernel and cannot be circumvented by applications. Type Enforcement is used to implement data flow structures called Assured Pipelines. Assured pipelines are made possible by the so-called "small process" model of computation used by Unix. In this model, a computational task is divided up into small virtual units that run in parallel to each other. Unix provides a crude and loosely-controlled way of sharing data between processes. Type Enforcement supplants this with the rigorously controlled, configurable structure of assured pipelines.

In addition, secure computer **80** has been configured under BSD386 to run in one of two states: administrative and operational. In the administrative state all network connections are disabled and the Server will only accept commands from a properly authenticated System Administrator accessing the system from the hard-wired administrative terminal (such as terminal or workstation **50** in FIG. 2). This feature prevents anyone other than the System Administrator from altering the security databases in secure computer **80**.

In the operational state the network connections are enabled and the Server will execute only software which has been compiled and installed as executable by an assured party.

The two states are reflected in two separate kernels. The administrative kernel is not subject to Type Enforcement. Instead, it is network isolated and accessible only to authorized personnel. This means that in administrative kernel mode, secure computer **80** cannot be seeded with malicious software by any but the people charged with system administration.

On the other hand, the operational kernel is subject to Type Enforcement. This means, for instance, that executable files stored in the memory of secure computer **80** cannot be executed without explicit execution privileges. In one such embodiment, executable files cannot be given execution privileges from within the operational kernel. Instead, secure computer **80** must enter administrative kernel to grant execution privileges. This prevents execution of malicious software posted to secure computer **80** memory. Instead, only executables approved by operational administrators while in administrative kernel mode ever become executable within operational kernel mode of secure computer **80**. In such an embodiment, administrative kernel can be entered only from either a manual interrupt of the boot process to boot the administrative kernel or by booting secure computer **80** from a floppy that has a pointer to the administrative kernel.

These restrictions provide the following advantages:

Defense in Depth: If an attacker should find a vulnerability in a system **40** subsystem, the damage that attacker can cause is limited to that subsystem. This prevents well-known attacks where a vulnerability in, e.g., the mail subsystem can be exploited to take over an entire installation.

Silent Alarms: The Type Enforcement supersedes and constrains the traditional "root" and "superuser" privileges of insecure Unix. Attempts to exercise these privileges in system **40**, or to violate other constraints of Type Enforcement, result in alarms being raised in administrative processes. No signal or indication of attack detection need be given, however. Instead, system **40** can, if desired, gather data to trace the source of the attack, feed false or misleading data to the attackers or take other appropriate countermeasures.

Open Security Architecture: The modular design means new Internet services can be provided quickly and securely.

An example of an assured pipeline appears in the diagram shown in FIG. *5a*. The flow of data between processes in FIG. *5a* is controlled by the access enforcement mechanism of the Intel Pentium processor. Virtual memory translation circuitry within the Pentium processor includes a mechanism for assigning access privileges to pages of virtual memory. This ensures that control is imposed on every fetch from, or store to, the machine memory. In this way, the protection is made continuous. The Pentium access control mechanism enforces the following modes of access:

Read Only (R): Data values may be fetched from memory and used as inputs to operations, but may not be modified or used as program text.

Read Execute (RE): Data values may be fetched from memory and used as inputs to operations, and may also be used as program text, but may not be modified.

Read Write (RW): Data values can be fetched from memory and used as inputs to operations, and may also be stored back in modified form.

No Access: The data cannot be fetched from memory for any purpose, and it may not be modified.

The diagram in FIG. *5a* then shows how these hardware-enforced accesses are used to force data flowing from internal network **82** to the Internet to go through a filter process, without any possibility that the filter is bypassed or that filtered data is tampered with by possibly vulnerable software on the Internet side of the filter.

The access a process has to a data object via Type Enforcement is defined by an entry in a central, protected data structure called the Domain Definition Table (DDT). A representative DDT is shown in FIG. *5b*. A Domain name denotes an equivalence class of processes. Every process in

execution has associated with it two Domain names which are used to control its interaction with object and with other Domains. The real Domain of a process is used to control Domain to Domain interactions and to grant or deny special, object-independent privileges. The effective Domain of a process is used to control its access to objects. The real and effective Domains of a process will generally be identical; the circumstances in which they differ are described below.

A Type name denotes an equivalence class of objects. Objects are, in general, the "base types" of BSD/386 Unix: files, directories, etc. There are eight default subtypes: file, directory, socket, fifo, device, port, executable, and gate. The implied default subtype pipe is, in effect, untyped because no check is made on access to pipes. The source code names of these subtypes are given in the table in FIG. 6.

Type names consist of two parts and, in the preferred embodiment, are written in documentation and comments as creator:subtype. The creator field is the four-character name of the Domain which created the object. The subtype field denotes the "class" of the object within that Domain. Subtype names are also four characters long and may contain any printable character except '*' or whitespace.

The TESLA convention is that subtypes will not be shared; thus Mail: file means, in effect, "the files private to the Mail Domain." When object are created they are automatically assigned the appropriate default subtype. Objects which are to be shared between Domains must have their subtype changed from the default to an explicit subtype.

Subtypes can be assigned one of three ways:

By having a default subtype assigned when the object is created by the operational kernel.

By having an explicit subtype assigned by the privileged chtype or fchtype syscalls. Thus a file which was to be shared between the Mail Domain and some other Domain would first be created as Mail:file and then changed to, e.g., Type Mail:Publ. If an subtype is changed to a default subtype, then the object becomes private.

By having a default or explicit subtype assigned administratively by the administrative kernel.

The default subtypes exec and gate are "static." The operational kernel will not create any objects of those subtypes, change those subtypes into any other subtype, or change any other subtypes into a gate or exec.

The Domain/Type relationship is used to define the modes and consequences of accesses by processes to objects. The modes and consequences of accesses are defined by access attributes which are store in the DDT database. The DDT database is "indexed" by three values:

The effective Domain of the process requesting the access or action.

The creator field of the object Type.

The subtype field of the object Type.

The result of "indexing" is the retrieval of a set of access attributes. The term "attribute" is used instead of "mode" because some of the attributes define immediate side effects. The selection of attributes was governed by the following considerations.

To constrain the modes of access which processes may exercise on objects.

To prevent he execution of any application software other than that which has been installed through the controlled administrative environment.

To enable the spoofing of attackers so that the attack response facilities can be used to trace them at the physical packet level. This required a more sophisticated response to illegal accesses than just shutting down the offending process.

The possible access attributes and their meanings are given in the table in FIG. 7.

Interactions Between Domains and Domains

The rules which govern the setting of the real and effective Domains of a process are as follows:

- Processes which are created by a fork syscall have their real and effective Domains set to the real and effective Domains of the parent process.
- If the executable used by execve syscall is of subtype exec, the real and effective Domains of the process are unchanged.
- The makedomain syscall may be used to change the real Domain of a process at the same time the executable is changed (analogous to execve). The new real Domain must be allowed by the DIT (the process is as shown in FIG. 11), and the effective Domain is changed to the new real Domain.
- the changedomain syscall may be used to change the real Domain of a process without changing the executable.
- if the executable used by execve is of subtype gate, the effective Domain of the process is set to the creator field of the full Type name of the executable. This action is called implicit gating. The new effective Domain must be allowed by the DIT.
- The gate syscall may be used to change the effective Domain of a process without changing the executable. The new effective Domain must be allowed by the DIT. This action is called explicit gating.
- The ungate syscall may be used to change the effective Domain of a process back to its real Domain. This action is called ungating.

Consider the case where a process running in the Mail Domain has execute access to files of Type Mail:exec and SMTP:gate. Further assume that there exists a Domain MIME. Then the new and effective Domains resulting from the relevant syscalls are shown in the table in FIG. 8. Gating facilities are not absolutely necessary for Type Enforcement to work. They exist for the following reasons:

- To simplify the DDT, by reducing the number of Types that would have to exist simply to implement inter-Domain data flow.
- To improve performance, by reducing the amount of copying and signalling required to coordinate activities in different Domains.
- To facilitate the porting of existing code whose process structure was not determined or influenced by considerations of least privilege or confinement of effect.

Gating permits a process to temporarily become a member of another Domain. The "home" or permanent Domain of the process is called its real Domain and the temporary or assumed Domain is called the effective Domain.

Implicit gating is used when it is necessary to strictly control the manner in which the effective Domain's accesses are used. Implicit gating "ties" the temporary Domain change to a specific executable which has been subjected to extra scrutiny to insure that the effective Domain's accesses are used safely. The "tying" of the Domain change is done because the Domain change is a side effect of execve'ing a special executable: one whose subtype is gate. Implicit gating also allows Domain changes to be defined by changing the Type of an executable instead of inserting explicit calls into the source code.

Explicit gating is used when a looser control on the temporary Domain transition is appropriate, or when the "tying" of the gating to a specific executable would require excessive restructuring of existing software.

Domain changes are controlled by the DIT. The logical structure of the DIT is a table with an entry for each Domain. The logical structure of each entry is that of two pointers, one to a list of allowed real Domains and the other to a list of allowed effective Domains. Thus, if a process executed a makedomain or changedomain, the real Domain of the process selects the entry and the Domain given by the domainname argument must be on the list of allowed real Domains for the Domain change to happen. Likewise, if a process executes a gate, the Domain given in the domainname argument must be on the list of allowed effective Domains. Finally, if a process executes an execve of an executable whose subtype is gate, the creator Domain of that executable must appear on the list of allowed effective Domains.

Certain kernel syscalls are restricted to processes executing out of privileged Domains. The specific restrictions are given below. The privileges which may be granted to a Domain are listed in the table in FIG. 9. The DDT matrix consists of columns indexed by data class and rows indexed by process class. The diagram in FIG. 5*b* shows the DDT elements used to enforce the assured pipeline shown in FIG. 5*a*. Whenever a data access is initiated, as through a Unix open call, the DDT is consulted and the memory management hardware of the Pentium processor is initialized with the defined access value. After that, the enforcement is done transparently and with no performance impact by the hardware. Other Unix systems calls are treated slightly differently, but the principle is the same.

In the preferred embodiment of Type Enforcement two levels of checks are made. First the normal BSD UNIX permissions are checked; if these permissions cause the operation to fail, the system call returns the normal error code. If the UNIX permissions are adequate the TE privileges are check next. (and thus in addition to the UNIX permissions).

The following BSD system calls have been modified to properly implement Type Enforcement. The modified calls have been grouped into four groups for ease of explanation.

The first group of system calls that require modification are those that set or affect the identity and/or state of the computer. Two of these system calls affect the computer's internal time: settimeofday and adjtime. Both of these system calls have been modified to require the <can_set_clock> privilege before the request will be honored. In the event of a privilege violation, the system call will raise an Alarm, will not honor the request, but will return success.

Other system calls which affect the computer's notion of self identity are sethostname and sethostid. Both of these system calls have been modified to require the <is-startup> privilege before the request will be honored. In the event of a privilege violation, the system call will raise an Alarm, will not honor the request, and will return the EPERM error flag. The last system all affects the computers runtime status, reboot. The reboot system call has been modified to require the <admin-reboot> privilege before the request will be honored. If the request is honored, the computer will boot to the admin kernel (single-user mode only with networking disabled). In the event of a privilege violation, the system call will raise an Alarm, will not honor the request, and will return the EPERM error flag.

The second group of system calls the require modification are those that allow interaction with the computer's filesystem. The open system call has been modified to become the primary TE check. After performing the normal BSD UNIX permission checks, the TE check is performed. An Alarm is raised if the TE check returns null (no permissions), or if the caller asks for read but the <ddt_read> privilege is not set, or if the caller asks for write but the <ddt_write> privilege is not set. The creat system call has been modified to set the new file's Type to <creator:file>. Additionally, the creation of a new file implies a write operation on the directory, which in turn implies that the TE-modified open system call will be used to open the directory file, which in turn implies that TE can be used to control the success or failure of the creat system call. The unlink and rename system calls are modified in like manner. The unlink system call requires the <ddt__destroy> privilege. The rename system call requires the <ddt__rename> privilege on the "from"file, and if the "to"file exists, it further requires the <ddt__destroy> privilege on the "to"file. In the event of a privilege violation, both the unlink and rename system calls will raise and Alarm, will not honor the request, but will return success. The access system call is modified to require the <mode> privilege on the file pointed to by the path. In the event of a privilege violation, the access system call will raise an Alarm, will not honor the request, but will return success. The chflags, fchflags and quotacl system calls are modified in alike manners. All are modified to perform no functions. Attempts to call them will raise an Alarm, will not honor the request, and will return EPERM. The mknod system call is modified to perform no function. Attempts to call it will raise an Alarm, will not honor the request, and will return EPERM.

The third group of system calls that require modification are those concerning process creation, maintenance and tracing. The fork system call has been modified so that the child process inherits both the real and effective Domains of the parent process. The execve system call is modified to require the <ddt__exec> privilege on the file pointed to by the path before the request will be honored. The real and effective Domain of the process remains unchanged. In the event of a privilege violation, the system call will raise an Alarm, will not honor the request, but will return success. The ktrace, ptrace and profil system calls are modified in alike manners. All are modified to perform no function. Attempts to call them will raise an Alarm and will not honor the request. The ktrace and ptrice system calls will return EPERM, whereas the profil system call will return EFAULT.

The mprotect system call is modified to perform no function. Attempts to call it will raise an Alarm, will not honor the request, and will return EPERM.

The fourth group of system call that require modification are those that relate processes to user ids. The setuid and seteuid and old.setreuid system calls are modified in alike manners. All are modified to require the <suppress__su__alarm> privilege before the request will be honored. In the event of a privilege violation, the system call will raise an Alarm, will not honor the request, and will return success. The acct system call is modified to perform no function. Attempts to call it will raise an Alarm, will not honor the request, and will return EPERM. The setlogin system call is modified to require the <can__setlogin> privilege. In the event of a privilege violation, the access system call will raise an Alarm, will not honor the request, but will return success.

A final set of system calls consists of those that are removed entirely from the BSD UNIX kernel. This set of system calls includes: obs__vtrace, nfssvc, asynch__daemon, getfh, shmsys, sfork, getdescriptor, and setdescriptor. (The set of system calls that were added to the BSD UNIX kernel is discussed elsewhere.)

The manner of searching the DDT is given in the diagram in FIG. 10. FIG. 10 shows the decomposition of a two-part Type name **100** into a creator domain D_C and a subtype T_S and the application of D_C and T_S to obtain permissions associated with the particular D_C and the particular T_S. Type name **100** is stored in the inode. In one embodiment, as is shown in FIG. **10**, D_C is used to obtain a pointer from DDT structure **104**. In one embodiment, DDT structure **104** is a C array of typedef permission__table. In one such embodiment, DDT structure **104** includes one element for each Domain in the system. In another embodiment, DDT structure **104** includes a wildcard entry * that matches all attempted selections in DDT structure **104**. (In the diagram shown in FIG. **10**, hard pointers are depicted as solid arrows while selectors are depicted as dashed line arrows.)

In one embodiment, as is shown in FIG. **10**, the pointer obtained from DDT structure **104** is used with T_S to obtain a pointer from subtype list **106**. In one embodiment, subtype list **106** is a C array of typed of ddt__type__list entry. In one such embodiment, subtype list **106** includes one element for each explicit subtype T_S that can be created by Domain D_C. In another embodiment, subtype list **106** includes a wildcard entry * that matches all attempted selections in list **106**.

In the embodiment shown in FIG. **10**, the pointer obtained from subtype list **106** is used in conjunction with effective domain D_E **102** to access Domain vector **108**. D_E **102** is the effective Domain of the program requesting access. If the program is executing an object of type D_O:exec, D_E comes from the process data structure. If the program is executing an object of type D_O:exec, D_E is set equal to D_O.

In one embodiment, Domain vector **108** is a C array of typed ddt__domain__vector__entry containing one element for each D_N that can use D_C. T_S. If a domain entry exists for D_E, access attributes **110** corresponding to D_E are returned. In one such embodiment, Domain vector **108** includes a wildcard entry * that matches all attempted selections of D_E in Domain vector **108**. The algorithm is as follows:

Obtain the Type name from the inode, where it is stored as a long, and parse it into two parts: the creator Domain D_C and the subtype name T_S.

Obtain the effective Domain **102**, D_E, from the process data base. If the executable object attempting the access is of Type D_G:gate, change D_E to D_O. (Note that a previous search of the DDT must have returned ddt__exec on the exec or gate object for this process to have begun.)

If $D_E = D_C$, and T_S is one of the default subtypes such as file (but not one of the "static" subtypes gate or exec) then return ddt__read+ddt__write+ddt__rename access.

If $D_E \neq D_C$, or if $D_E = D_C$ and T_S is not one of the default subtypes, then search the DDT structure **104** for the entry corresponding to D_C. If no such entry exists, search the structure for a "wildcard" entry. If neither an entry corresponding to DC, or a "wildcard entry" exists in the structure, assign null access.

If an entry for D_C exists, search the subtype list **106** it points to for an entry corresponding to T_S. If no such entry exists, search the subtype list it points to for a "wildcard subtype." If neither such entry exists, assign null access. If an entry for D_C does not exist, but a "wildcard" entry does, search the subtype list the "wildcard entry" points to for an entry corresponding to T_S. If no such entry exists, search the subtype list the "wildcard entry" points to for a "wildcard subtype." If neither an entry corresponding to T_S, nor a "wildcard subtype" exists in the subtype list, assign null access.

If a subtype list entry for T_S exists, search the Domain vector **108** it points to for an entry **110** corresponding to D_E. If no such entry exists, search the Domain vector for a "wildcard Domain." If neither an entry corresponding to DE, nor a "wildcard Domain" exists in the Domain vector, assign null access.

If a Domain vector entry for D_E exists, return the access values it contains. If a "wildcard Domain" entry exists in the Domain vector, return the access values it contains. If neither a Domain vector entry for D_E, nor a "wildcard Domain" exists in the Domain vector, return null access.

The above algorithm describes the "logical" process of searching the DDT; the actual implementation is described next.

As noted above, in one embodiment, Domains and subtypes are stored as four printable character constants (white space doesn't count as printable—also, '*' is excluded). Due to constraints imposed by the fact that BSDI Release 1.1 does not contain complete source code, only the first character of a Domain and the first three characters of a subtype are significant, and thus must be unique. Furthermore, there is a convention that subtype names that appear globally (i.e., both default subtypes and subtypes used by more than one Domain) be made of lowercase characters, while private subtypes be made of uppercase characters.

These four character names are represented by C constants. For Domains, these constants begin with a D, while for subtypes, these constants begin with a T. The following character should also be in uppercase (e.g., DRoot, TFile). There is also two special constants: kWildcard='****', which matches any subtype or Domain, and kEOL=0, which is used to mark the end of a list. These constants are all defined inside a list of enum's is since using #define would result in too many compiler warnings (the C compiler warns about multi-character constants, by using enum's, it will only warn once for a given constant).

There are six default subtypes, based on existing Unix types:

```
/* Here are the default types . . . */
TFile = 'file'
TDirectory = 'diry',
TSocket = 'sock',
TFifo = 'fifo',
TDevice = 'devi',
TPort = 'port',
TExec = 'exec',
TGate = 'gate'
};
```

TExec is a special subtype, which can only be assigned by the isolated administrative kernel. It represents executables which any Domain can execute if execute access is allowed by the DDT. TGate is a special sort of TExec—what it does is change the effective Domain in which a process is executing to the creator Domain of the gate. It only does this if the starting Domain has execute access to the file of subtype gate. After "gating," a process now acts like it is in the creator Domain for the purposes of the DDT checks only—any checks against the DIT are made with the real Domain, rather than the effective Domain. Needless to say, a gate is a powerful and potentially dangerous thing—just like the setuid bits which gating is designed to replace. Note that there is a special check in the normal DIT checks—if we are attempting to change to the real Domain, we don't bother to check the DIT of the effective Domain (like we otherwise normally would). This maneuver is "ungating"—explicitly leaving the gated Domain and returning to the original Domain.

There is only one pre-defined Domain:

```
enum {
      DRoot = '$SYS', /* Root is actually a special alias for the
                         zero domain that the system is started in */
};
```

which is used to represent system level defaults—whenever a Domain that hasn't been explicitly set (for either a file or a process), DRoot is used for the Domain value in permission checks.

The DDT is made of a three level table, indexed by the file's creator Domain, file's subtype and then finally by the executing Domain. This yields a set of access permissions:

```
typedef unsigned long ddt_permissions;
  enum {
        ddt_read = 1,
        ddt_write = 2,
        ddt_rename = 4,
        ddt_exec = 8,
        ddt_trigger = 0x10,
        ddt_chcreator = 0x20,
        ddt_destroy = 0x40,
        };
```

These permissions work mostly as expected—ddt_read, ddt_write are for read and write; ddt_rename permits changing the name of the file; ddt_exec is used to grant execute permission; ddt_destroy is required to delete a file. ddt_chcreator is much like create permission, but since files are created with a default subtype, this permission allows the given Domain to change the subtype and creator of the file to the corresponding subtype/creator pair. ddt_trigger isn't really a permission—rather, any checks to this specific file will automatically trigger an alarm, regardless of what permission is asked for or granted. This allows, for example, a "reverse trojan" file that would never be executed except by an attacker, in which case an alarm would be triggered and packet-level auditing performed.

The indexing begins with an array "indexed" by Domain:

```
typedef struct {
type_name src_domain; /* What domain this entry is for */
        unsigned long domain_flags; /* The "global" permission flags */
type_name * the_dit; /* What domains we can enter into */
        ddt_type_list the ddt; /* The permission for our types */
} permission_table;
```

This array should have an entry for every Domain. For the DDT, this table is searched until the src_domain matches the creator Domain of the file. Assuming that it is found, we then look at the,_ddt an array "indexed" by subtype:

```
    /* This is the permission for a specific domain, listing all its types */
typedef struct {
    type_name the_type; /* The subtype */
ddt_domain_vector_entry * the vector,/* A list of what can be done to it */
} ddt_type_list_entry, *ddt_type_list;
```

We just look through this list until we either find the subtype, a wildcard, or the end of the list (in which case we return no permission). We then need to look at the appropriate the_vector—an array "indexed" by Domain:

```
typdef struct {
     type_name the_domain; /* The using domain */
```

-continued

```
    ddt__permissions the permission; /* What it can do */
} ddt__domain__vector__entry, *ddt__domain__vector;
```

This is search for the executing Domain, and if found, we return the__permission, which contains the flags for that access.

Searching the DIT starts like searching the DDT. We look through the global table for the starting Domain, and find the appropriate list of Domains the__ddt. This is simply a list of Domains, terminated with kEOL. We search through that list, and if we find the desired destination Domain, we can make the transition to it.

Every Domain also has a list of privileges that it can perform:

```
enum {
        can__ch__type = 0x00010000, /* We can call ch__type, changing type */
        suppress__su__alarm = 0x00020000, /* Allow process to think it is su */
        admin__reboot = 0x00040000, /* Allow reboot */
        can__set__clock = 0x00080000 /* Can set the clock */
        can__setlogin = 0x00100000 /* Can perform setlogin */
        is__startup = 0x00200000 /* can perform startup actions */
```

We look through the permission table to find the appropriate Domain, and then get these permissions from the appropriate domain__flags field. Note that there is no explicit "can__ch__domain" permission; restrictions on Domain transitions are enforced by the DIT.

Since each and every array must be a separate C structure, every array needs to have a unique and meaningful name to connect one array to its parent. This is best explained in a "simple" example.

```
/* NB: In the initial implementation, domains need to have unique
        first characters */
enum    {
        DRoot = '$SYS', /* Root is actually a special alias for the zero
                                    domain that the system is started in */
        DUpdate = 'sync',
        DSwap = 'Swap'
        DUserSession = 'User',
        DSyslogd = 'Logd',
        DCron = 'Cron',
        DRouted = 'Rout',
        DSendmail = 'mail',
        DInetd = 'inet',
        DTelnet = 'inet',
        DShell = 'rshd',
        DRExec = 'exec',
        DFinger = 'fing',
        DNetwork = 'xnet',
        DLpd = 'lpd',
        DPortmap = 'port',
        DFsck = "Fsck',
        DQuota = 'quot',
        DXDos = 'dosx',
        DFtp = 'Tran',
        DInnd = 'News'
};
```

These are just a list of sample Domains:

```
type__name Root__dit[] = {
DUserSession, DSyslogd, DUpdate, DCron, DRouted, DLpd, DPortmap,
DSendmail, DInetd, Dinnd, kEOL
```

These are some addition private subtypes for our example:

```
        /* Here are some other types */
        enum {
                TStartup = 'Stup',
                TConfig = 'Conf',
                TCronJobs = 'CJob'
```

This is the list of Domains that the root Domain can change to. The naming convention here is DomainName__dit, where DomainName is the name of the constant for that Domain without the leading "D". The Domain list is terminated with a kEOL.

```
        { ddt__domain__vector__entry Root__Startup[] = {
        { kWildcard, ddt__read },
        { kEOL } };
```

This is our first Domain vector. The naming convention is CreatorDomainName__TypeName, where CreatorDomainName is the name of the constant for the creating Domain (without the leading "D"), and TypeName is the name of the constant of the subtype. Vector is initialized to contain a list of Domains and permission paris, terminated with {kEOL}.

```
ddt__domain__vector__entry Root__default [] = {
  /* This is the default permissions for all procs on all unassigned files */
        { kWildcard, ddt__read\ddt__write\ddt__rename },
        { kEOL }
};
```

Root__default will be the Domain vector for creator Root, and subtype kWildcard—basically the default for any subtypes created by DRoot who otherwise wouldn't have a special Domain vector.

```
ddt__domain__vector__entry Root__Exec[] = {/* Default for executables */
    { kWildcard, ddt__exec\ddt__read };
    { kEOL }
};
```

Another Domain vector, this time for all executables owned by the system.

```
ddt__type__list__entry Root__types [] = {
        { TStartup, Root__Startup },
        { TConfig, Root__Startup },
        { TExec, Root__Exec },
        { kWildcard, Root__dDefault },
        { kEOL }
};
```

Once we have all the Domain vectors for a given creating Domain, we can make the corresponding subtype list. The naming convention is CreatingDomain_types. It is composed of pairs of subtypes and the corresponding (previously declared) Domain vectors. Note that it is possible for more than one subtype to use the same Domain vector (in this case, both TStartup and TConfig).

```
permission_table Rober D  = (
     { DRoot,
           can_ch_type\can_ch_creator\admin_reboot\can_set_clock,
           Root_Dit, Root_Types };
     {DInetd,
          0,
         Inetd_dit, NULL }
,    { kEOL }
};
```

Here is the master permission table "Rover". It is composed of a list of Domains (two in this case). Each entry contains the Domain name, the permissions for that Domain, its DIT and its subtype list. If the DIT is NULL, then no Domain transitions out of that Domain are allowed. If its subtype list is NIL, then there is null access to all subtypes of that creating Domain. The last entry, of course is the kEOL termination.

Every process runs in a Domain, which is stored in the kernel proc data structure. This property is copied to processes that are forked, and is unchanged by executing most binaries and shell scripts. The Domain can be explicitly changed via the makedomain system call, which, if permitted, changes Domain for that process from that point forward. Privileges of a given Domain can also be granted to something running in another Domain via a "gating" process—a process that executes a file of subtype gate will, assuming there is execute permission granted to the current Domain for that file, temporarily assumes the privileges of the creator of the gate file. This is accomplished by an "effective Domain:" field in the kernel proc data structure. This field is also copied during forking, and is reset when makedomain is successfully called (reset to the new Domain specified). Most importantly, the effective Domain field is used to check file access permissions, but real Domain is used for checks from makedomain. There is, however, a special addition to makedomain for the purpose of "ungating"—if the process is calling makedomain with the real Domain, it automatically succeeds (thus resetting the effective Domain to the real Domain), allowing a process to return to the Domain that it started in. The Domain transition permissions are all handled in domain_to_domain. This routine first looks up the source Domain in the permissions table. It will use the kWildcard entry, if any, to provide default source Domain permissions. It then looks in the DIT vector for the destination Domain, and, if found, allows the transition. It will not, however, use a wildcard in that vector, since this would allow a given Domain to transition to every other Domain.

The most important check that execve makes is to check for ddt_exec access. It looks at the subtype and creator of that which is to be executed and the effective Domain of the current process (not the real Domain), and makes sure that there is ddt_exec access. If there is it also compares the subtype of the file to see if it is gate—if so, we change the effective Domain to the creator of that file.

There is also logic in execve that makes sure that we don't gate by mistake—the old effective Domain is grabbed at the start of execve, and any time that an error is returned, we first restore the old effective Domain.

chtype/fchtype are used to change the subtype and/or creator of a file. Because of this power, they must be carefully controlled. One of the first constraints on chtype is that it can either change both the subtype or creator. We can never change anything about a file that we aren't currently the creator of. Furthermore, since exec and gate are special static subtypes, we can never make or unmake an exec or gate. This is only done from the administrative kernel. The final special rule is that we can only change to a subtype/creator that already exists (this is to prevent making "orphaned" object, but with the special kWildcard type we could still specify access permissions for these things, so this rule could be removed). Note that this "check for existence" will accept wildcards in the permission table as matching whatever we pass in.

The other checks made by chtype/fchtype are checks to the permission table. First off, the executing Domain needs to have can_ch_type permission. Then, if we are only changing the subtype of an object that we created (and all the checks in the previous paragraph pass), then we just go ahead and do that. If however, we are changing the creator as well, we check the ddt to see if our effective Domain (as opposed to real Domain—see gates for more detail) has chcreator capabilities for the creator/subtype that we are going to change the file to (we already know that we created it, so we don't care what the subtype is). If we do then we change it, if not, then we don't.

Actually changing the subtype, since we are hacking subtype and creator into the flags field of the vnode, requires us to be running as root (since we are changing both words of the flags field, and VOP_SETATTR seems to care). So, before calling VOP_SETATTR, we first save the cr_uid, set it to zero, and then restore it. When we modify VOP_SEATTR to write our subtype and creator to the real places in the inode, this will be removed.

check_ddt takes an effective Domain and a creator:subtype pair and looks for specific access attributes, returning those that correspond to permissions, and raising alarms if things don't work as expected. The first thing that check_ddt does, after mapping any potentially undefined fields to DRoot and/or TFile (if the subtype or creator is zero, such as on a file system not properly set up), is check for the default subtypes. If the source Domain is the same as the creator, and the subtype is one of the default eight subtypes, the returned access attributes are ddt_read+ddt_write+ddt rename.

Otherwise, we need to look up the creator.subtype in our tables. If we find them (or appropriate wildcard matches), we then search the Domain vector to find the source Domain. If we find that (or again, the wildcard), the return permission is taken from there. If we never find one of the respective entries, the return permission is no permission.

The last step in check_ddt is to see if the return attribute is inconsistent with the permissions asked for by the caller, or if the resulting permission includes the ddt_trigger attribute. If either of these cases are true, then we need to log this request to the alarm mechanism. This involves writing out the process id, the name of the file, the parameters and what permission is returned. The alarm processing would, at that point, take appropriate action.

In addition, the system **40** shown in FIG. **4** is constructed so that no software may be loaded into it except under the control of the System Administrator, and even then only when the system is disconnected from all networks. (This is a function of the two kernels: operational and administrative, as described above.)

The Type Enforcement mechanism allows a strict least privilege design to be defined and enforced. Least privilege is a way of achieving confinement, or the limiting of a software module's effects. A least privilege design is one in which software only touches the data it needs to operate and only touches it in ways that the designer intended. Unwanted side effects, whether from bugs or malicious trojan horses, are then limited to the module's "immediate vicinity." This fundamental ability of Type Enforcement, when properly applied, stops dead the most common types of attacks, where a vulnerability in one application is used to interfere with, or take control of, more critical sections of the system.

In order to take advantage of this capability, the application only needs to follow traditional Unix practices and be implemented as several processes. These processes can be assigned to a distinct class, as can the data that they access. The DDT can be configured to allow only the least amount of access necessary for the desired functionality.

The Type Enforcement described above permits a security architect to construct a set of interconnected applications and protect them with countermeasures such as data filters. The architect can do this with the confidence that the applications and countermeasures will be isolated from each other and share data only in the ways the architect defines. This enables the architect to upgrade system **40** quickly to respond to changes in threat, by adopting new countermeasures; to secure new applications, by constructing countermeasures that address the specific vulnerabilities of the application; and to implement customer-specific security policies which balance risk against operational effectiveness.

Since Type Enforcement defines pipelines and subsystems which are independent with regard to privilege, the addition of a new subsystem or the extension of a pipeline does not, in and of itself, obsolete the assurance evidence produced for the previous structure. Rather, the assurance team can examine the new interactions and decide precisely which conclusions about isolation are still valid and which have to be re-examined.

Type Enforcement has also demonstrated its ability to support cryptography, whether implemented in hardware and software. Cryptographic processing, with its requirements for separation of plaintext and ciphertext, is inherently a pipelined process. This is true whether the cryptography is placed in its traditional "inline" position or whether it is used in the "coprocessor" mode required for the more advanced services such as digital signatures and non-repudiation.

Type Enforcement is better than the basic Unix protection mechanisms for two reasons: it is centralized instead of decentralized, and it does not permit any process to have global, uncontrolled access. In Unix, individual programs use the setuid mechanism to set their own privilege level. A particular privilege level, called "root," or "super-user," lets a user do anything they want to the system: observe and manipulate data, disable auditing, install trojan horses, or masquerade as other users. This combination of decentralization and potential global privilege is deadly. Decentralization means that there is no one place you can look to see

if the system is configured securely. Global privilege means that a single vulnerability or configuration mistake can be catastrophic.

Type Enforcement eliminates both these problems. If you stop a system **40** as described in FIG. **4** and dump the DDT you can tell for sure which code could ever have touched which data. You can never tell that in a Unix system. And nobody ever gets global privilege when secure computer **80** is attached to a network.

In the preferred embodiment, the Type Enforcement restrictions supplement, but do not replace, the standard Unix permissions. That is, you can set Unix permissions to give less, but not more, access than Type Enforcement allows. And super-user privilege is still there, but it cannot be used to exceed the Type Enforcement limitations.

In one embodiment, a system **40** detects an attack in progress (as a result, for instance, of a Type Enforcement violation) it trips a "silent alarm" which is responded to by application-specific countermeasure software. This software can, depending on the nature of the attack, do the following things:

Capture the IP address of the attacking site, enabling calls to site administrators to trap attackers in the act.

Feed the attacker false and misleading data.

Feed the attacker useless but "interesting" data so he stays on-line and can be traced.

Feed the attacker data containing covert identification data that can be used to prove that data was stolen from this site.

In one embodiment, a binary filter is used to ensure that neither executables nor encrypted files are transferred into or out of system **40**. (The prohibition against executables is an attempt to capture malicious software transferred into the system and to detect the posting of potentially proprietary object code from system **40** onto the Internet. The prohibition against transfer of encrypted files is an attempt to prevent the posting of encrypted versions of proprietary information either to or from system **40**.) In one binary filter embodiment, text is analyzed to determine if it is written in English. The filter looks at each character and its next neighbor and determines the frequencies of pairs of letters ("a diagraphic index of correlation"). If the index of correlation approximates what would be expected for English text, the file is probably English text and can be transferred. If not, filter **92** stops the transfer. One embodiment of a binary filter command which could be used advantageously in the present invention is listed in Appendix A.

Operation of the Secure Wide-Area Access System

When a Client desires to put information out on Public Network **74**, he or she must first use the Local Cryptography to establish and authenticated and protected interaction with Secure Computer **48**. The Client then issues the requisite commands through the Client interface, and these commands and their associated are then executed and controlled by the integrated set of services and filter counter-measures on the Secure Computer. The Public Network Protocol and Cryptography module then selects the appropriate authentication and protection mechanism for the interaction on Public Network **74**. Depending on the protocols and cryptography used, Public Network **74** and Cryptography module **70** may then perform cryptographic and format transformations on the data. Most commonly, these would involve decrypting data that was encrypted using Local Cryptography, changing its format from a local messaging or data transfer format to a global standard, and encrypting using Global Cryptography. At the same time, Secure Computer **48** can generate an audit record and protect it with cryptographic keying material accessible only to authorized administrators.

If authentication is required, Secure Computer 48 can either "endorse" or "notarize" the data using cryptographic keying material of its own, or it can act as a secure storage and selection facility whereby the local authentication of the Client is used to select the personal keying material used to authenticate the Client's identity on Public Network 74. Secure Computer 48's facilities can use other information, such as time of day, content of the data, etc., as well as the facilities of the Local Cryptography to decide whether or not to perform authentication of the outbound information.

An important special case is where two systems **40** at two different sites belong to the same organization. Such a situation is shown in FIGS. **12** and **13**. In FIG. **12**, two systems **40** are connected by an external Public Network 74. In FIG. 13, two systems **40** connected by an external Public Network 74 can also communicate with an unclassified workgroup **100** or with individual computers **102** and **104** connected directly to Network 74. In such cases, special protocols and keying material can be used to identify the systems to each other and indicate special actions, such as administrative changes and alarms. In addition, systems **40** can easily distribute keys between themselves in a secure manner. In one embodiment, systems **40** include Trusted Path software which can be used to establish a trusted path between independent systems **40** over Public Network 74.

Inbound information flow is essentially symmetric to outbound: the data is received from Public Network 74, if necessary decrypted and has its authentication checked, and then is passed through the Filter Countermeasures 68 to determine whether the organizational security policy allows data of that label, format, or content to be released into Private Network 64. If it does, Secure Computer 48 uses Local Cryptography to protect and authenticate the transmission to Client Workstation 63. When the Client accesses the data, he or she cain use that cryptography to verify that the data is what it was authenticated to be over the Public Network 74.

Advantages Over Other Methods of Securing Data Transfer

The general advantages of the invention derive from its centralization of security services in Secure Computer 48. This centralization takes advantage of the fact that Client workstations 63 must be supported by centralized services such as directories for electronic mail, databases of security attributes, and archival storage of cryptographic keys. Thus every Security Architecture which makes use of cryptography is, to one degree or another, centralized.

Similarly, the facilities for detecting and responding to security alarms are most usefully centralized. Notifying a Client in a possibly exposed location that a network is possibly under attack can be counterproductive: the Client may not be authorized for such information, and even if authorized the individual may not have a secure means of communicating this information to administrators. Also, one does not want to notify a possible insider threat that an attack has been detected. Thus again a degree of centralization in the architecture is unavoidable. Further centralization of security mechanisms adds both security and economic benefits:

1) Mechanisms at the workstations can be implemented as software and minimal, if any hardware. This implementation strategy limits the strength of the workstation mechanisms, and is only acceptable when they are "backed up" by the strength and facilities of a central Secure Computer and the restricted access inherent in a Private Network.

2) Concentration of the security requirements and facilities in the Secure Computer enables that unit to undergo

scrutiny to a degree that would not be feasible for individual workstations. If the Secure Computer is properly engineered it should be able to support multiple generations of workstation technology, thereby spreading the cost of specialized security engineering over time.

3) Concentration of countermeasures in a specially-engineered Secure Computer raises the effort and risk of technical attacks because it forces the attacker to either reverse engineer and implement, or obtain through other means an up-to-date copy of the Computer and all its associated countermeasures software. This is harder than obtaining an instance of a workstation and its associated software. Concentration also simplifies the process of responding to new or unanticipated attacks, as there are fewer units to change and those units are already under the control of security administrators.

4) Concentration also simplifies the process of administering the security databases and increases the speed and reliability with which privileges can be granted and, more importantly, revoked.

5) The Secure Computer will, by its very nature, have the features which make it a near-optimum platform for key management and distribution: strong authentication of individuals, secure storage of data, controls on access to that data and strong resistance to attacks by malicious software.

6) The Secure Computer, by virtue of its central role and close interaction with security administrators, provides a logical and effective location for the receipt and response to security alarms. This characteristic combines with the ability to respond to new attacks by upgrading a smaller number of central sites and the speed and effectiveness of changes to security data bases to make the centralized approach inherently more responsive than architectures without a central point of security enforcement, where alarms, changes to software, and changes to data bases must propagate over a larger number of user-administered workstations.

In particular, the invention provides superior client authentication over methods such as Workstation Cryptography. In Workstation Cryptography, Clients authenticate themselves at vulnerable workstations by means of personal identifiers such as passwords, passphrases, Personal Identification Numbers, or token-based authenticators. There is no protected backup or contextual check possible on such authentication actions; once authenticated, the Client is granted, in effect, full access to the Public Network. By contrast, the Secure Computer can keep a protected record of Client actions, and assess the propriety of an authenticated action based on that data as well as other criteria such as time of day, whether it is a business day or a holiday, or other checks of arbitrary sophistication. Conversely, the invention permits the sending of "official" data or transactions in which the identity of the initiating individual is shielded from the Public Network and only the organizational identity is authenticated. This facility is useful when the nature of the transaction or data could make the Client open to unwanted attention, harassment, or retaliation.

The invention provides an advantage over Workstation Cryptography in that it is possible to enforce sophisticated, content-based organizational security policies. Such enforcement is not possible when data is enciphered at the workstation and then sent directly to the Public Network. In addition to enforcing content-based policies, the invention permits auditing of data contents to deter abuse of the privilege of sending data to the Public Network. Both of these facilities are useful in countering insider threats.

The invention is superior to Workstation Cryptography in that it can handle a multitude of communications protocols and cryptographic methods without making that diversity visible at the Client workstation. This not only reduces the amount of hardware and software mechanism at the multiple workstations, but it permits a single Client Interface to be used to access a heterogeneous Public Network. The Secure Computer, after it has decrypted data that was protected and authenticated by the Local Cryptography, can consult internal tables, directories on the Public Network, or the destination node to determine or negotiate a common protocol and cryptographic method. All of this can be done without Client knowledge or intervention.

The invention is superior to Workstation Cryptography in that it provides a safer and more reliable framework for the management of keying material. This advantage obtains irrespective of whether secret-key or public-key cryptography is applied. The Secure Computer provides a central site for the distribution and administration of keying material for all the Clients on the Private Network, and relieves the Client workstations of the responsibility of obtaining Public Network keying material for every interaction with that network. The distribution of Public Network keying material through the Secure Computer permits greater security in that the identities of the requesting Clients can be hidden from the Public Network keying material service. The invention also provides superior solutions to the problems of revocation, emergency rekey, and travelling user.

The use of the Secure Computer as the central point for the distribution and administration of keying material permits the effective and efficient revocation of access to either the Private or the Public Networks. In the most common configuration, secret-key methods will be used by local Cryptography and public-key methods will be required for Global Cryptography. If the private key of a Client's public-key material are distributed to Client workstations, or, worse, stored on removable tokens that the Client can remove, then revocation of the ability to decrypt (or, more importantly, authenticate) data requires a time-consuming and unreliable "broadcast" of revocation requests to all possible destinations on the Public Network. If the private key is kept on the Secure Computer, then access to it can be revoked simply and quickly.

The invention is superior to Workstation Cryptography in providing emergency rekey service, especially when public-key methods tire used on the Public Network. If the private key part of a Client's public-key material is lost or destroyed, the Client loses the ability to decrypt data which was previously encrypted with the corresponding public key. It is not sufficient to issue a new private/public pair, because there may be data in transit or in archives that was enciphered with the public key that corresponds to the lost private key. The problem then is one of saving a copy of the private key in a highly protected fashion, and making it available only after proper authorization has been obtained. This is a natural task for a Secure Computer with protected storage and mechanisms and access limited to authorized administrators. If the organization has Secure Internetwork Services Systems at multiple sites, then they can cooperate by maintaining backup copies of critical keying material for each other.

The invention is superior to Workstation Cryptography in that a Secure Computer at one site can forward the necessary keying material to another site, whether it be a Secure Internet Services System or some other node on the Public Network. This forwarding can be closely controlled and audited, and the superior revocation facilities used to place a limit on the period during which the forwarded material can be used.

The invention is superior to Network Cryptography in that it permits controls, auditing, protection, and authentication to the granularity of the individual Client rather than just to the node.

Although the present invention has been described with reference to the preferred embodiments, those skilled in the art will recognize that changes may be made in form and detail without departing from the spirit and scope of the invention.

APPENDIX A

5.2 "Nobinary" Filter

5.2 1 Overview

The No binary filter checks a text file for files that are unrecorded or base-64 encoded or any other form other than English plain-text. This is done by counting digraphs in the body of the message, performing some statistical calculations on the number of digraphs and comparing the results to previously calculated statistical properties of english plain-text.

5.22 Options

Most options can be set at the command line with the exception of the list of files that were discovered by the filter. The place for this list is set in the h file, but output is only generated if the -l command line switch is given.

5.2.3 Compile Time (defines in nobin.h)

BADFILE_LIST: Define where the list of rejected files will be put

5.2.4 Command line Switches

-R -r

Descend through the directory hierarchy

-a

Check all printable ASCII characters, default is to check alphabetic characters only.

-s

Tells the filter not to skip headers in files to be examined.

-l

Tells the filter to generate a listing of the files it found to be binaries.

-c

Tells the filter to calculate the sum of the squares of the probabilities of all characters checked for. Will print to stdout.

-p

Prints results of each file to stdout.

-f <filesize__cutoff>

The Filter will ignore all files with a size less than <filesize__cutoff> (in bytes) default is 2000.

-i <ic__cutoff>

The Filter will mark all files with an I.C. less than <ic__cutoff>as binaries. The default is the I.C. between 1.00 (Random Text) and 1/(size of character setÔ2), which is 4.6644 when only checking alphabetic characters and is 15.7355199 when checking all printable ASCII characters

5.2.5 Notes

Since this filter will run after the standard news programs, it assumes it will not be looking at any binaries. If it runs on a binary, it will probably get an IC of infinity and not tag it as an ASCII-encoded binary because it is only checking for printable ASCII characters.

All files should take on the standard news/mail format. If you wish to check files other than-news/mail format, use the -s switch to turn off the skip headers flag.

If desired, the filter can scan a directory hierarchy by using the -R flag. For example. nobin -R/usr/spool/news/comp/binaries will scan the comp. binaries. * news hierarchy for posts that are other than English-language plaintext. Without the -R switch, it will just look at/usr/spool/news/comp/binaries and see that it is directory and quit.

What to do with the rejected files has yet to be decided. There is a procedure called DoBinFile( ) that has been set aside but as of now (01 Aug. 1994) it does nothing.

What is claimed is:

1. A system for secure internetwork communication across an external network, the system comprising:

first and second internal networks;

first and second secure computers connected to the external network, wherein said first and second secure computers are type enforcing secure computers capable of recognizing data of varying sensitivity and of limiting access to data based on both user access rights and process access rights and wherein the first and second secure computers include:

an internal network interface; and

an external network interface for secure transfer of data from the first secure computer to the second secure computer over the external network, wherein the external network interface includes means for encrypting data to be transferred from the first secure computer to the second secure computer;

a first computing system, wherein the first computing system includes a first client subsystem connected over the first internal network to the internal network interface of the first secure computer, wherein the first client subsystem includes means for secure transfer of data between the first computing system and the first secure computer; and

a second computing system, wherein the second computing system includes a second client subsystem connected over the second internal network to the internal network interface of the second secure computer, wherein the second client subsystem includes means for secure transfer of data between the second computing system and the second secure computer.

2. The system according to claim 1 wherein the first secure computer further comprises:

means for selectively filtering messages received from the second internal network according to a first predefined criteria; and

means for selectively filtering data received from the external network according to a second predefined criteria.

3. A secure computing system, comprising:

an internal network;

an external network;

first and second secure computers connected across the external network, wherein the first and second secure computers comprise encryption means for encrypting and decrypting data transferred between said first and second secure computers and wherein the first secure computer further comprises means for establishing an assured pipeline between processes operating on said internal network and processes operating on said external network; and

a workstation connected across the internal network to said first secure computer, wherein the workstation includes means for encrypting and decrypting data transferred between said workstation and said first secure computer.

4. The system according to claim 3 wherein the first secure computer further comprises:

means for selectively filtering messages received from the internal network according to a first predefined criteria; and

means for selectively filtering data received from the external network according to a second predefined criteria.

5. The system according to claim 3 wherein the first secure computer is a multilevel secure computer capable of recognizing data of varying sensitivity and users of varying authorizations.

6. The system according to claim 3 wherein the first secure computer is a type enforcing secure computer capable of recognizing data of varying sensitivity and of limiting access to data based on both user access rights and process access rights.

7. A method of transferring data between a first and a second network connected by an external network, wherein the first network comprises a first workstation connected to a first secure computer server and wherein the second network comprises a second workstation connected to a second secure computer server, wherein each secure computer server comprises a trusted subsystem, first encryption means for encrypting and decrypting data transferred between the secure computer server and its respective workstation and second encryption means for encrypting and decrypting data transferred between the secure computer server and the external network, the method comprising the steps of:

establishing an authenticated and protected interaction between the first workstation and the first secure computer server;

establishing an assured pipeline between processes operating on said first network and processes operating on said external network;

sending data from the first workstation to the first secure computer server;

passing the data received from the first workstation through the assured pipeline;

selecting an authentication and protection mechanism for interaction on the external network;

encrypting, via the second encryption means of the first secure computer server, the data received from the first workstation through the assured pipeline; and

sending the encrypted data over the external network to the second secure computer server.

8. A secure server, comprising:

a processor;

an internal network interface, connected to the processor for communicating on an internal network; and

an external network interface, connected to the processor for communicating on an external network;

wherein the processor includes server program code for transferring data between the internal and external network interfaces via an assured pipeline and security policy program code for enforcing a Type Enforcement security mechanism to restrict access of a process received from the external network to data stored on the internal network.

9. The server according to claim 8, wherein the processor further includes encryption means for encrypting data to be transferred from the internal network to the external network.

10. The server according to claim 8, wherein the processor further includes filter program code for filtering data transferred between the internal and external network interfaces.

11. The server according to claim 8, wherein the processor further includes formatting program code for changing the format of data transferred between the internal and external network interfaces.

12. A secure server for use in controlling access to data stored within an internal network, comprising:

administrative and operational kernels, wherein the operational kernel includes security policy program code for enforcing a Type Enforcement security mechanism to restrict access of a process received from the external network to data stored on the internal network; and

wherein the administrative kernel is restricted to execution only while isolated from the internal network.

13. A secure wide-area access system, comprising:

a secure computer;

an internal network; and

a workstation connected across the internal network to the secure computer;

wherein the secure computer comprises an internal network interface, a public network interface, public network program code used to communicate through the public network interface to a public network, private network program code used to communicate through the internal network interface to the workstation and security policy program code for enforcing a Type Enforcement security mechanism to restrict access of a process to data.

14. The system according to claim 13 wherein the security policy program code comprises program code for hardening an operating system.

15. The system according to claim 13 wherein the security policy program code comprises program code for hardening an operating system, wherein the program code for hardening an operating system comprises kernel code for enforcing Type Enforcement via an operational kernel.

16. A method of protecting a computer system connected to an unsecured external network, wherein the computer system includes a plurality of workstations connected to a private network, the method comprising the steps of:

providing a secure computer, wherein the secure computer comprises security policy program code for enforcing a Type Enforcement security mechanism to restrict access of a process to data;

connecting the Type Enforcement based secure computer to the private network; and

establishing an assured pipeline for the transfer of data and programs between the private network and the external network through the secure computer, wherein the step of establishing an assured pipeline includes the steps of:

i) placing processes within domains, wherein the step of placing processes within domains includes the step of assigning processes received from the external network to an external domain;

ii) assigning types to files; and

iii) restricting access by processes within the external domain to certain file types.

17. The method according to claim 16 wherein the step of placing processes within domains includes the steps of:

defining a domain definition table within the secure computer;

assigning a domain name to each domain; and

creating an entry for each process in the domain definition table, wherein the step of creating an entry includes the step of associating a domain name with each entry.

18. A system for transferring data between a workstation connected to an internal network and a remote computer connected to an external network, the system comprising:

a secure computer, wherein the secure computer includes:

an internal network interface connected to the internal network; and

an external network interface connected to the external network, wherein the external network interface includes means for encrypting data to be transferred from the workstation to the remote computer;

means for establishing an assured pipeline between said internal network interface and said external network interface;

a server function for transferring data between the internal network interface and the external network interface, wherein the server function includes means for establishing an assured pipeline between said internal network interface and said external network interface; and

filter means for filtering data transferred between the remote computer and the workstation.

19. The system according to claim 18, wherein the workstation includes a client subsystem for secure transfer of data from the workstation to the internal network interface of the secure computer.

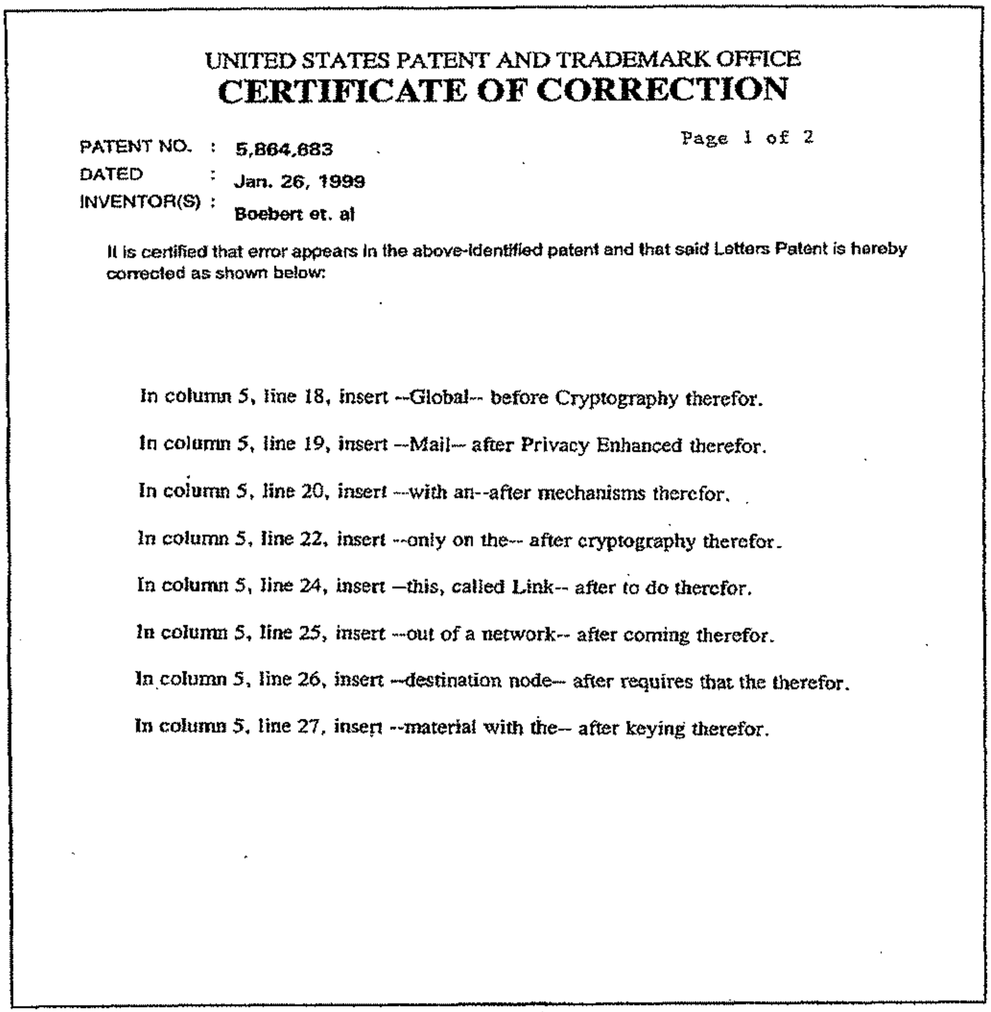
20. The system according to claim 19, wherein the client subsystem is a software program running on the workstation.

21. The system according to claim 20, wherein the secure computer further comprises:

means for selectively filtering messages received from the internal network according to a first predefined criteria; and

means for selectively filtering data received from the external network according to a second predefined criteria.

* * * * *



UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

Page 1 of 2

PATENT NO.	:	5,864,683
DATED	:	Jan. 26, 1999
INVENTOR(S)	:	Boebert et. al

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In column 5, line 18, insert --Global-- before Cryptography therefor.

In column 5, line 19, insert --Mail-- after Privacy Enhanced therefor.

In column 5, line 20, insert --with an--after mechanisms therefor.

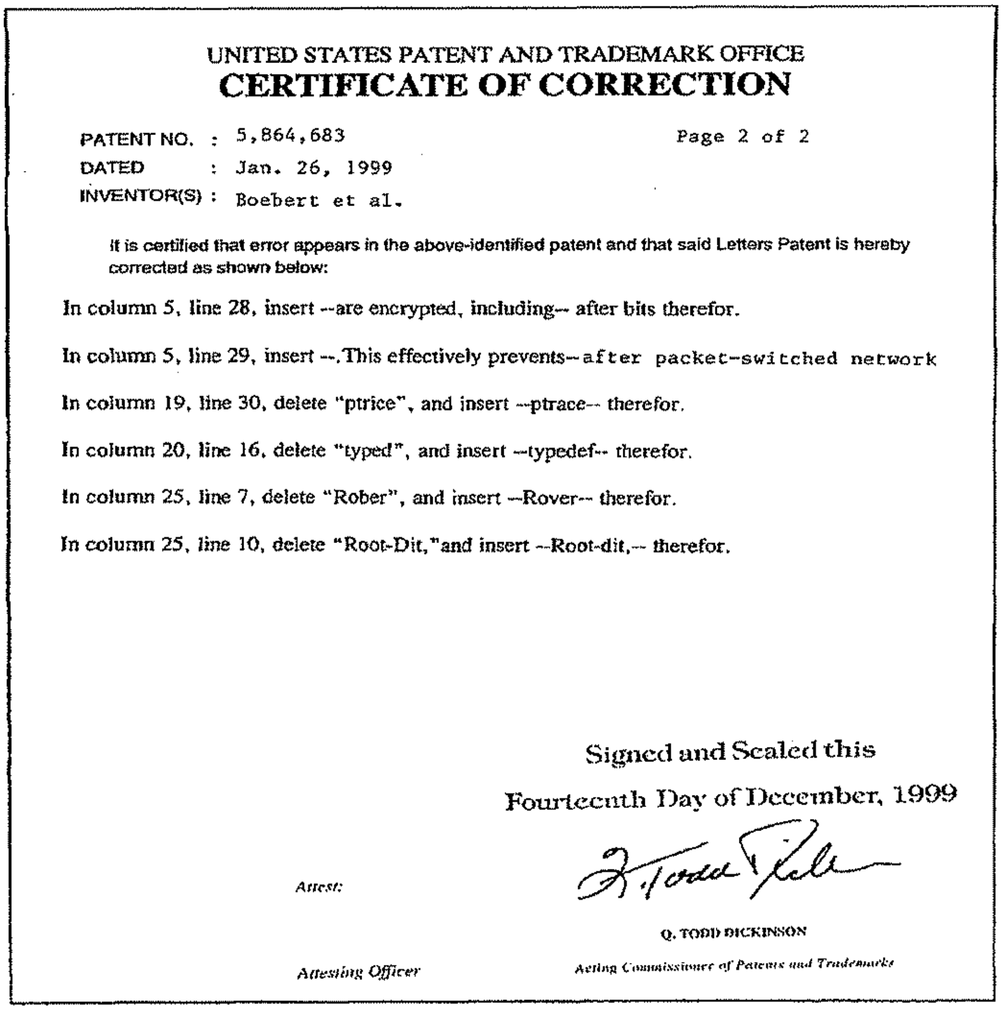
In column 5, line 22, insert --only on the-- after cryptography therefor.

In column 5, line 24, insert --this, called Link-- after to do therefor.

In column 5, line 25, insert --out of a network-- after coming therefor.

In column 5, line 26, insert --destination node-- after requires that the therefor.

In column 5, line 27, insert --material with the-- after keying therefor.



UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 5,864,683 Page 2 of 2
DATED : Jan. 26, 1999
INVENTOR(S) : Boebert et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In column 5, line 28, insert --are encrypted, including-- after bits therefor.

In column 5, line 29, insert --.This effectively prevents--after packet-switched network

In column 19, line 30, delete "ptrice", and insert --ptrace-- therefor.

In column 20, line 16, delete "typed", and insert --typedef-- therefor.

In column 25, line 7, delete "Rober", and insert --Rover-- therefor.

In column 25, line 10, delete "Root-Dit,"and insert --Root-dit,-- therefor.

Signed and Sealed this
Fourteenth Day of December, 1999

Attest:

Q. Todd

Q. TODD DICKINSON

Attesting Officer *Acting Commissioner of Patents and Trademarks*

CURRENT PROJECTS System

THE NATIONAL ACADEMIES
Advisers to the Nation on Science, Engineering, and Medicine

Search FullText Search

Navigation Menu
- Home
- Search for Projects
- View Projects
- by Project Title
- by Subject/Focus Area
- by Board/Committee
- by Major Unit
- Provisional Committee Appointments Open for Formal Public Comment
- by Last Update
- Meeting Information
- Conflict of Interest Policy
- Committee Appointment Process
- FAQ

More Project Information and to provide FEEDBACK on the Project

Printer Friendly Version

Committee Membership Information

Project Title: Technical and Privacy Dimensions of Information for Terrorism Prevention and Other National Goals

PIN: LJXX-I-04-02-A

Major Unit: Division on Behavioral and Social Sciences and Education

Sub Unit: Committee on Law and Justice
Committee on National Statistics
Computer Science and Telecommunications Board

RSO: Chemers, Betty

Subject/Focus Area:

Committee Membership
Date Posted: 07/03/2006

Dr. Charles M. Vest - (Co-Chair)
Massachusetts Institute of Technology

Charles M. Vest is the president emeritus of MIT. He chairs the U.S. Department of Energy Task Force on the Future of Science Programs and serves as vice chair of the Council on Competitiveness. He is also a member of both the Center for Strategic and International Studies (CSIS) Commission on Scientific Communication and National Security and of the National Academies-CSIS collaborative Roundtable on Scientific Communication and National Security. He is a member of the Executive Committee of the Association of American Universities (AAU) and recently completed terms as AAU chair and vice chair. Dr. Vest has been a member of the President's Council of Advisors on Science and Technology since 1994. Prior to assuming the MIT presidency in 1990, Dr. Vest was Provost and Vice President for Academic Affairs at the University of Michigan, where he previously served as Dean of Engineering. He is a director of IBM and E.I. du Pont de Nemours & Company; a trustee of the University Corporation for Advanced Internet Development; a member of the Corporation of the Woods Hole Oceanographic Institution; and an ex officio institutional trustee of the WGBH Educational Foundation. He is chair of the advisory board of TIAX, a founding member of the Board of Associates of the Whitehead Institute for Biomedical Research, and a member of the board of directors of the Blanchette Rockefeller Neurosciences Institute. He recently served on the U.S. Commission to Combat Proliferation of Weapons of Mass Destruction.

Mr. Fred H. Cate
Indiana University

Fred H. Cate is a Distinguished Professor at the Indiana University School of Law--Bloomington and director of the Indiana University Center for Applied Cybersecurity Research. He served as counsel to the Department of Defense Technology and Privacy Advisory Committee, was a member of the Federal Trade Commission?s Advisory Committee on Online Access and Security, and directed the Electronic Information Privacy and Commerce Study for the Brookings Institution. He is currently a senior policy advisor to the Center for Information Policy Leadership at Hunton & Williams, an Indiana law firm, a member of Microsoft?s Trustworthy Computing Academic Advisory Board, and a member of the board of editors of Privacy and Information Law Report. He is actively involved with information privacy and security issues outside of the United States. He chaired the International Telecommunication Union?s High-Level Experts on Electronic Signatures and Certification Authorities, served as a member of the United Nations Working Group on Emergency Telecommunications and has advised the governments of China, Finland, Japan and Thailand. Professor Cate is the author of many articles and books, including Privacy in the Information Age, Privacy in Perspective, and The Internet and the First Amendment. He is an elected member of the American Law Institute. His research interests are in information law issues, particularly in the context of digital networks. His teaching interests include communication law, electronic commerce and communications, internet and privacy law. He received his J.D. and A.B. in History with Honors and Distinction from Stanford University.

Dr. Ruth A. David
ANSER (Analytic Services Inc.)

Ruth A. David is the president and chief executive officer of ANSER, a not-for-profit, public-service research institution that provides research and analytic support on issues relating to international and domestic terrorist threats. Dr. David is a member of the Department of Homeland Security Advisory Council (HSAC), the National Academy of Engineering (NAE) and the Corporation for the Charles Stark Draper Laboratory, Inc. She is vice chair of the HSAC Senior Advisory Committee of Academia and Policy Research and serves on the National Security Agency Advisory Board, the National Research Council Naval Studies Board, the NAE Committee on Engineering Education, the American Association for the Advancement of Science Committee on Scientific Freedom and Responsibility, the Jet Propulsion Laboratory?s Technical Division Advisory Board, and the External Advisory Committee for Purdue?s University Homeland Security Institute. From September 1995 to September 1998, Dr. David was Deputy Director for Science and Technology at the Central Intelligence Agency. As Technical

Advisor to the Director of Central Intelligence, she was responsible for research, development, and deployment of technologies in support of all phases of the intelligence process. She represented the CIA on numerous national committees and advisory bodies, including the National Science and Technology Council and the Committee on National Security. Prior to moving to the CIA, she was Director of Advance Information Technologies at Sandia National Laboratories where she began her professional career. She is the recipient of many awards including the CIA?s Distinguished Intelligence Medal, the CIA Director?s Award, and the Director of NSA Distinguished Service Medal. She is a former adjunct professor at the University of New Mexico. Her research interests include digital and microprocessor-based system design, digital signal analysis, adaptive signal analysis and system integration. Dr. David received her Ph.D, in Electrical Engineering from Stanford University.

Dr. Ruth M. Davis
Pymatuning Group, Inc.

Ruth M. Davis is president and chief executive officer of the Pymatuning Group, Inc. in Alexandria, Virginia, which specializes in industrial modernization strategies and technology development. She serves on the boards of 12 corporations and private organizations and was a member of the board of regents of the National Library of Medicine from 1989 to 1992. She is a former chairman of the Aerospace Corporation and served as assistant secretary of energy for resource applications, and deputy undersecretary of defense for research and advanced technology. She has taught at Harvard University and at the University of Pennsylvania, and she currently serves on the University of Pennsylvania?s board of overseers of the School of Engineering and Applied Science. She also serves on a number of advisory committees to the federal government and is on the Council of the National Academy of Engineering. She was elected to the National Academy of Engineering in 1976. She has served on more than two dozen NAS panels and committees. She has a Ph.D. in mathematics from the University of Maryland. Her research interests include expediting the development process for law enforcement technologies and she has worked extensively on means of identifying meaningful requirements for law enforcement technologies and ensuring adequacy of life cycle functions. She has studied and written on the technical and managerial features of the technology based threat to information assets.

Dr. William H. DuMouchel
Lincoln Technologies, Inc.

William H. DuMouchel is Vice-President, Research, and the Chief Statistical Scientist at Lincoln Technologies, Inc., Wellesley Hills, Massachusetts. Previously, Dr. DuMouchel was a senior statistician at AT&T Labs-Research and a member of Lincoln Technologies? Board of Directors and a senior advisor to the company. Dr. DuMouchel is the inventor of the empirical Bayesian data mining algorithm known as GPS and its successor MGPS, which have been applied to the detection of safety signals in databases of spontaneous adverse event reports. These methods are now used within the FDA and industry. From 1987-1992, he was a Chief Statistical Scientist at BBN Software Products, where he was lead designer of a software advisory system for data analysis and experimental design called RS/Discover and RS/Explore. Dr. DuMouchel has been on the faculties of the University of California at Berkeley, the University of Michigan, MIT, and most recently was Professor of Biostatistics and Medical Informatics at Columbia University from 1994?1966. He has also been an associate editor of the Journal of the American Statistical Association, Statistics in Medicine, Statistics and Computing, and the Journal of Computational and Graphical Statistics. Dr. DuMouchel is an elected fellow of the American Statistical Association and of the Institute of Mathematical Statistics, and has served previously on the National Research Council Committee on Applied and Theoretical Statistics. Most recently he served on the IOM Committee on Postmarket Surveillance of Medical Devices. He received a Ph.D. in Statistics from Yale University. His research focuses on statistical computing, Bayesian hierarchical models, including applications to meta-analysis and data mining.

Dr. Stephen E. Fienberg
Carnegie Mellon University

Stephen E. Fienberg is the Maurice Falk University Professor of Statistics and Social Science at Carnegie Mellon University. He is an editorial board member of the Journal of Quantitative Criminology, Philosophia Mathematica, and Research of Official Statistics and co-editor of the Section for Statistics of the International Encyclopedia of the Social and Behavioral Sciences. He served for seven years as a Member, DBASSE, National Research Council, as the president of the Institute of Mathematical Statistics, and president of the International Society for Bayesian Analysis. Dr. Fienberg is the author. co-author or editor of numerous books including Discrete Multivariate Analysis: Theory and Practice; The Analysis of Cross-classified Categorical Data, Statistics and the Law; Intelligence, Genes, and Success: Scientists Respond to the Bell Curve; and Who Counts? The Politics of Census-Taking in Contemporary America. He has participated in numerous National Academies committees and workshops, the most recent being Committee to Review the Scientific Evidence on the Polygraph (2003) and the Committee on the Review of the National Immunization Program?s Research Procedures and Data Sharing Program (2005). Dr. Fienberg?s current research interests include Bayesian approaches to confidentiality and data disclosure; causation; foundations of statistical inference; sample surveys and randomized experiments; statistics and the law; and inference for multi-media data. He holds as Ph.D. in Statistics from Harvard University.

The Honorable William J. Perry - (Co-Chair)
Stanford University

William J. Perry is the Michael and Barbara Berberian Professor at Stanford University, with a joint appointment at the Stanford Institute for International Studies (SIIS) and the School of Engineering. He is also a senior fellow at SIIS and the Hoover Institution, and serves as co-director of the Preventive Defense Project, a research collaboration of Stanford and Harvard Universities. He was the co-director of the Center for International Security and Arms Control (CISAC) from 1988 to 1993, during which time he was also a half time professor at Stanford. Dr. Perry was the 19th secretary of defense for the United States, serving from February 1994 to January 1997. He previously served as deputy secretary of defense (1993-1994) and as under secretary of defense for research and engineering (1977-1981). Perry is on the board of directors of several emerging high-tech companies and is chairman of Global Technology Partners. His previous business experience includes serving as a laboratory director for General Telephone and Electronics (1954-1964); founder and president of ESL Inc. (1964-1977); executive vice- president of Hambrecht & Quist Inc (1981-1985); and founder and chairman of Technology Strategies & Alliances (1985-1993). He is an expert in U.S. foreign policy, national security and arms control. He is a member of the National Academy of Engineering and a fellow of the American Academy of Arts and Sciences. He received a B.S. and M.S. from Stanford University and a Ph.D. from Pennsylvania State University, all in mathematics.

Mr. W. Earl Boebert
Sandia National Laboratories

W. Earl Boebert is an expert on information security, with experience in national security and intelligence as well as commercial applications. He is a senior scientist at Sandia National Laboratories. He has 30 years experience in communications and computer security, is the holder or co-holder of 13 patents, and has participated in CSTB studies on security matters. Prior to joining Sandia, he was the technical founder and chief scientist of Secure Computing Corporation, where he developed the Sidewinder security server, a system which currently protects several thousand sites. Before that he worked 22 years at Honeywell, rising to the position of senior research fellow. At Honeywell he worked on secure systems, cryptographic devices, flight software, a variety of real-time simulation and control systems, and won Honeywell's highest award for technical achievement for his part in developing a very large scale radar landmass simulator. He also developed and presented a course on systems engineering and project management that was eventually given to over 3,000 students in 13 countries. He served on the CSTB committees that produced Computers at Risk: Computing in the Information Age; For the Record: Protecting Electronic Health Information; and Information Technology for Counterterrorism: Immediate Actions and Future Possibilities. He also participated in CSTB's workshops on "Cyber-Attack" and "Insider Threat."

Dr. Michael L. Brodie

Verizon Communications

Michael L. Brodie is chief scientist of Verizon IT (information technology). Dr. Brodie works on large-scale strategic information technology challenges for Verizon Communications Corporation's senior executives. His primary interest is in the optimal use of IT, with an emphasis on emerging and advanced technologies and practices, to enable organizational and business objectives, including organizational change. He also investigates the relationships between economics, business, and technology, and computing-communications convergence. His long-term industrial research focus is on advanced computational models and architectures and the large-scale information systems that they support. He is concerned with the Big Picture, business and technical contexts, core technologies, and "integration" within a large scale, operational telecommunications environment. Dr. Brodie has authored over 150 books, chapters, journal articles, and conference papers. He has presented keynote talks, invited lectures, and short courses on many topics in over thirty countries. He is a member of the boards of six research foundations including the VLDB (Very Large Databases) Endowment (1992 – 2004); the Advisory Board of the School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne, Switzerland (2001 – present); Advisory Board, Digital Enterprise Research Institute, National University of Ireland (2003-present); expert advisor to the Information Society Technologies priority of the European Commission's Sixth and Seventh Framework Programmes (2003-present); and is on the editorial board of several research journals. He received his Ph.D. in computer science from the University of Toronto in 1978.

Mr. Duncan A. Brown
Johns Hopkins University, Applied Physics Laboratory

Duncan A. Brown is a member of the Principal Staff and Director of the Strategic Assessments Office (SAO) at the Johns Hopkins University Applied Physics Laboratory (JHU/APL) located in Laurel, Maryland. The SAO conducts broad-ranging analyses and assessments of National Security strategy, policy and technology trends that may affect APL. Recent efforts have included conducting an alternative futures exercise to examine potential geopolitical strategic futures and their impact on the military and related R&D, conducting an effort for OSD and the Secretary of the Navy's Office to examine the principles of war, providing technical analysis and advice to the Defense Advanced Research Projects Agency, and serving on a National Reconnaissance Office and Navy sponsored study panel to assess the future use of space. Prior efforts have included Submarine Force wartime readiness assessments, creation of the U.S. Navy's Unmanned Combat Aerial Vehicle Program, serving on a Naval Research and Advisory Committee Panel to examine issues associated with transitioning technology, and serving on a Naval Studies Board to examine the role of experimentation in building future Naval Forces. Mr. Brown has also served on the Navy staff in the Pentagon as the Science Advisor to the Deputy Chief of Naval Operations, in the Pacific as the Science Advisor to the Commander in Chief Pacific Fleet, and in the Pentagon as the Director for Submarine Technology. Mr. Brown also headed the Hydrodynamics Branch at the Naval Undersea Warfare Center in Newport, R.I. The Branch was responsible for investigating drag and noise reduction techniques for submersibles using both numerical simulations as well as test models in its own tow tanks and on ranges. Mr. Brown has received three Navy Superior Civilian Service Awards. Mr. Brown's formal education includes graduate work in National Security Studies at Georgetown and MIT. He was a Fellow in MIT's Seminar XXI Foreign Politics and International Relations in the National Interest Program. Mr. Brown also holds an M.S. degree from Johns Hopkins University in Engineering Management, a M.S. degree in Ocean Engineering from the University of Rhode Island, and a B.S. degree in Engineering Science from Hofstra University.

Dr. Cynthia Dwork
Microsoft Research

Cynthia Dwork has been a senior researcher at Microsoft Research at its Silicon Valley Campus since 2001. Previous positions include staff fellow, Compaq Systems Research Center (2000-2001 and research staff member at IBM Almaden Research Center (1985—June 2000). Since 1997, she has served as a consulting professor at Stanford University. Most of Dwork's research has been in cryptography and other topics in distributed computing. She is the co-inventor of non-malleable cryptography and of the only public-key cryptosystem for which random instances are probably as hard to break as the hardest instances of the underlying mathematical problem. Her research interests include foundations of cryptography, database privacy, complexity theory, web search, distributed computing, interconnection networks, algorithm design and analysis. She serves on the editorial boards of Journal of Algorithms, Information and Computation; Journal of Cryptology, Internet Mathematics; and the Chicago Journal of Theoretical Computer Science. Dr. Dwork received her B.S.E. from Princeton University; and her Ph.D. in computer science from Cornell University.

Dr. Robert J. Hermann
Global Technology Partners, LLC

Robert Hermann is senior partner of Global Technology Partners, LLC, which specializes in investments in technology, defense, aerospace and related businesses worldwide. In 1998, Dr. Hermann retired from United Technologies Corporation where he held the position of senior vice- president, science and technology. In this role, he was responsible for assuring the development of the company's technical resources and the full exploitation of science and technology by the corporation. He was also responsible for the United Technologies Research Center (UTRC). Dr. Hermann joined the company in 1982 as vice-president, systems technology, in the electronics sector and later served in a series of assignments in the defense and space systems groups prior to being named vice-president, science and technology. Prior to joining UTRC, he served 20 years with the National Security Agency with assignments in research and development, operations and NATO. In 1977, he was appointed principal deputy assistant secretary of defense for communications, command, control and intelligence. In 1979, he was named assistant secretary of the Air Force for research, development and logistics and in parallel was director of the National Reconnaissance Office. He received B.S., M.S. and Ph.D degrees in electrical engineering from Iowa State University.

Mr. R. Gil Kerlikowske
City of Seattle, Washington

R. Gil Kerlikowske is a 32-year law enforcement veteran, and was appointed as the chief of police for the Seattle Police Department on August 14, 2000. He was the former deputy director for the U.S. Department of Justice, Office of Community Oriented Policing Services that provides federal grants to local police agencies in support of community policing services. He served as the police commissioner for Buffalo, New York, where his selection by the mayor became the first outside appointment in 30 years. Kerlikowske also served as the chief of police for two Florida cities, Fort Pierce and Port St. Lucie, both of which received the Attorney General's Crime Prevention Award. In 1985 he was a visiting fellow at the National Institute of Justice where he designed an evaluation of police procedures throughout the country. He began his law enforcement career in 1972 as a police officer for the St. Petersburg Police in Florida. Kerlikowske also served in the U.S. Army Military Police. He holds B.A. and M.A. degrees in criminal justice from the University of South Florida in Tampa, and is a graduate of the National Executive Institute at the Federal Bureau of Investigations Academy in Quantico, Virginia.

Mr. Orin S. Kerr
The George Washington University

Orin S. Kerr is associate professor at the George Washington University of School of Law. He is a prolific scholar in the area of criminal law and criminal procedure, and is nationally recognized as a leading voice in the emerging field of computer crime law. Kerr's recent scholarship has appeared in the Harvard Law Review, Columbia Law Review, Michigan Law Review, New York University Law Review, Georgetown Law Journal, Northwestern University Law Review, Hastings Law Journal, George Washington Law Review, William and Mary Law Review, Washington and Lee Law Review, and several other journals. His scholarship and advocacy in the field of Internet surveillance law has been profiled in the New York Times and National Public Radio's All Things Considered. From 1998 to 2001, Professor Kerr was an Honors Program trial attorney in the Computer Crime and Intellectual Property Section of the Criminal Division at the U.S. Department of Justice. He is also a former law clerk for Judge

Leonard I. Garth of the U.S. Court of Appeals for the Third Circuit and Justice Anthony M. Kennedy of the United States Supreme Court. Kerr received his B.S.E. in mechanical and aerospace engineering from Princeton University (1993); M.S. in mechanical engineering from Stanford University (1994); and J.D. from Harvard Law School (1997).

COMMUNICATIONS AND INFORMATION SYSTEMS

Dr. Robert W. Levenson
University of California, Berkeley

Robert W. Levenson is a professor in the Department of Psychology at the University of California, Berkeley and is the director of the Institute of Personality and Social Research and the Berkeley Psychophysiology Laboratory. He has published numerous papers on the autonomic nervous system: including "Autonomic nervous system distinguishes among emotions," "Emotion and the autonomic nervous system: A prospectus for research on autonomic specificity," "Emotion and autonomic nervous system activity in the Minangkabau of West Sumatra," and "Hiding feelings: The acute effects of inhibiting negative and positive emotion." His research interests include the physiological, facial expressive and subjective aspects of emotion, and the emotional changes in neurodegenerative disorders and depression. He has served as a board member for the American Psychological Association, co-chair of the Behavioral Sciences Workgroup at the National Institute of Mental Health and president of the Society for Psychophysiological Research. Dr. Levenson received his B.A. in psychology from Georgetown University and Ph.D. from Vanderbilt in community psychology.

Dr. Tom M. Mitchell
Carnegie Mellon University

Tom M. Mitchell is the Fredkin Professor of Computer Science at Carnegie Mellon University (CMU). He is president of the American Association of Artificial Intelligence, and author of the textbook Machine Learning. He is the founding director of CMU's Center for Automated Learning and Discovery, an interdisciplinary research center specializing in statistical machine learning and data mining. During 1999 to 2000, he served as chief scientist and vice-president for WhizBang Labs, a company that employed machine learning to extract information from the web. His research interests lie in the theory and application of machine learning algorithms, data mining, and cognitive science. His recent work has focused on machine learning approaches to text classification, information extraction, medical outcomes analysis, and analyzing human brain function based on MRI data. He currently serves on the Computer Science and Telecommunication Board (CSTB). He also served on the CSTB committee that produced the report Information Technology for Counterterrorism: Immediate Actions and Future Possibilities. In 2004, he became the chair-elect, American Association for the Advancement of

Science, Section on Information, Computing, and Communication. Dr. Mitchell received his Ph.D. in electrical engineering with a computer science minor from Stanford University.

Dr. Tara O'Toole
Johns Hopkins Bloomberg School of Public Health

Tara O'Toole is the chief executive office and director of the Center for Bio-security at the University of Pittsburgh Medical Center, and professor of medicine at the University of Pittsburgh. O'Toole was one of the original members of the Johns Hopkins Center for Civilian Bio-defense Strategies, and served as director of the Hopkins Center from 2001 to 2003. She was one of the principal authors and producers of "Dark Winter," an influential exercise conducted in June 2001 to alert national leaders to the dangers of bioterrorist attacks. From 1993 to 1997, O'Toole served as assistant secretary of energy for environment safety and health. As assistant secretary, Dr. O'Toole was the principal advisor to the secretary of energy on matters pertaining to protecting the environment and worker and public health from the U.S. nuclear weapons complex and DOE laboratories. From 1989 to 1993, Dr. O'Toole was a senior analyst at the Congressional Office of Technology Assessment (OTA), where she directed and participated in studies of health impacts on workers and the public due to environmental pollution resulting from nuclear weapons production. Dr. O'Toole is a Board-certified internist and occupational medicine physician. She received her bachelor's degree from Vassar College, her M.D. from the George Washington University, and an M.PH. from Johns Hopkins University. She completed a residency in internal medicine at Yale, and a fellowship in occupational and environmental medicine at Johns Hopkins University.

Dr. Daryl Pregibon
Google, Inc.

Daryl Pregibon is the research scientist at Google, Inc. He is a recognized leader in data mining, the interdisciplinary field that combines statistics, artificial intelligence, and data base research. His research interests include analysis of massive data sets, statistical computing, generalized linear models, tree-based methods, and regression diagnostics. During his career, Dr. Pregibon has nurtured successful interactions in fiber and microelectronics manufacturing, network reliability, customer satisfaction, fraud detection, targeted marketing, and regulatory statistics. Over these years, his research contributions changed from mathematical statistics to computational statistics and included such topics as expert systems for data analysis, data visualization, application-specific data structures for statistics, and large-scale data analysis. From 1989-2004, he worked at AT&T and served as head, statistics research. He is currently a member of the NAS Committee on National Statistics; the NAS Study Committee on Ballistics and former chair of the NAS Committee on Applied & Theoretical Statistics. He has also held positions on the National Advisory Committee for the Statistical and Applied Mathematical Sciences Institute (SAMSI), Research Triangle Park and is director of the Association for Computer Machinery (ACM) Special Interest Group on Knowledge Development and Data Mining (SIGKDD). Other previous academic and professional experiences include: associate editor of Data Mining & Knowledge Discovery; associate editor, Statistics & Computing; and co-founder of the Society for Artificial Intelligence & Statistics (SAIAS). He has authored more than 60 publications and holds four patents. Dr. Pregibon received his Ph.D. in statistics from the University of Toronto and his M.A. in mathematics from Youngstown State University.

Dr. Louise Richardson
Harvard University

Louise Richardson is executive dean and senior administrative officer of the Radcliffe Institute for Advanced Study and is responsible for the coordination of academic and administrative activities and the strategic management of administrative operations. Richardson is also senior lecturer in government at the Faculty of Arts and Sciences at Harvard and lecturer in law at Harvard Law School. From 1989 to 2001, she taught as assistant and then associate professor of government at Harvard, specializing in international security. In July of 2001, Richardson was appointed executive dean of the Radcliffe Institute for Advanced Study. Richardson's academic focus has been on international security with an emphasis on terrorist movements. In addition to her three books: What Terrorists Want, Democracy and Counterterrorism: Lessons from the Past and When Allies Differ: Anglo-American Relations in the Suez and Falkland Crises, Richardson has published a number of journal articles, book chapters, and reviews on the subject of terrorism. These include "Five Degrees of Separation: Terrorists and their Sponsors," "Terrorists as Transnational Actors," "A Spiral of Peace? Bringing an End to Ethnic Violence in Northern Ireland,""Ending Terrorist Campaigns: Lessons from War Termination," "To Escalate or Not to Escalate, That is the Question: Factors Driving Terrorist Decisions to Escalate," and "Conflict Theory and Terrorist Campaigns." Richardson's current research involves a study of decision-making inside terrorist movements and a study of the patterns of terrorist violence. Dean Richardson received a bachelor's degree in history from Trinity College, Dublin. She earned M.A's in political sciences from UCLA and Harvard and in history from Trinity College, Dublin, and a Ph.D. in government from Harvard University.

Dr. Ben A. Shneiderman
University of Maryland, College Park

Ben Shneiderman is a professor in the Department of Computer Science, founding director (1983-2000) of the Human-Computer Interaction Laboratory, and member of the Institute for Advanced Computer Studies and the Institute for Systems Research, all at the University of Maryland at College Park. He has taught previously at the State University of New York (SUNY) and at Indiana University. He was made a fellow of the Association for Computing Machinery (ACM) in 1997, elected a fellow of the

American Association for the Advancement of Science in 2001, and received the ACM CHI (Computer Human Interaction) Lifetime Achievement Award in 2001. In addition he has co-authored two textbooks, edited three technical books, and published more than 200 technical papers and book chapters. In 1999 he co-authored Readings in Information Visualization: Using Vision to Think with Stu Card and Jock Mackinlay, and in 2003 continued in this direction by co-authoring The Craft of Information Visualization: Readings and Reflections with Ben Bederson. Ben Shneiderman's vision of the future is presented in his October 2002 book Leonardo's Laptop: Human Needs and the New Computing Technologies, which won the IEEE 2003 award for Distinguished Literary Contribution. He has consulted and lectured for many organizations including Apple, AT&T, Citicorp, GE, Honeywell, IBM, Intel, Library of Congress, Microsoft, NASA, NCR, and university research groups. He received his Ph.D. from State University of New York (SUNY) at Stony Brook.

Mr. Danny J. Weitzner
Massachusetts Institute of Technology

Danny Weitzner is the director of the World Wide Web Consortium's (W3C's) Technology and Society activities. As such, he is responsible for the development of technology standards that enable the web to address social, legal, and public policy concerns such as privacy, free speech, protection of minors, authentications, intellectual property and identification. He is also the W3C's chief liaison to public policy communities around the world and a member of the Internet Corporation for Assigned Names and Numbers Protocol Supporting Organization Protocol Council. Mr. Weitzner holds a research appointment at the Massachusetts Institute of Technology's Laboratory for Computer Science where he also teaches public policy. Before joining the W3C, he was a co-founder and deputy director of the Center for Democracy and Technology, an Internet civil liberties organization in Washington, D.C. He was also deputy policy director of the Electronic Frontier Foundation. His publications on communications policy have appeared in Yale Law Review, Global Networks, Computerworld, Wired Magazine, Social Research, Electronic Networking: Research, Applications and Policy, and The Whole Earth. Mr. Weitzner has a degree in law from Buffalo Law School and a B.A. in philosophy from Swarthmore College

Committee Membership Roster Comments
Note (07/03/06): There has been a change in committee members with the appointments of Duncan A. Brown and Orin S. Kerr.